
Use of Multimedia in Engineering Education

PhD Thesis

Richard Farr

Abstract

The work described in this thesis was conducted in order to investigate an apparent shortfall between what had been expected of multimedia in education, and what had been delivered. Despite the advantages attributed to multimedia and Computer Aided Learning (CAL) by the computing industry, there remained a shortage of suitable titles in some subject areas, including engineering.

Investigation revealed that the most significant barrier to the exploitation of multimedia technology concerned justification and payback for the substantial amount of development effort required to produce software of this kind. It was found that the size of the potential audience for a programme was all too easily limited by the exorbitant computer system requirements and limited flexibility which tended to be built into the software by default. It was aimed to investigate whether the elements of a multimedia programme which contributed greatly to its computer system requirements, cost and inflexibility were so closely linked to its educational effectiveness.

The research was experimental in nature. It involved the creation of several pieces of multimedia software, this being an experiment in itself since it allowed measurement of the amount of effort required to incorporate the various media into an educational programme. Two particularly significant pieces of software are described in detail in the thesis; an advisory system meant to promote design for testability among electronic engineers, and a CAL system offering an introduction to process planning. Both of these featured, in places, a highly interactive style, involving the dynamic generation of images and animations in response to users' input. This represented a radical departure from the conventional approach to multimedia, which was normally based upon the sequential playback of prerecorded material.

The process planning software was used with groups of students; their comments were invited and their performance was measured in a test which used a novel method to identify any students who had prior knowledge of the subject. (Correct answers from such people could not reasonably be claimed to indicate that learning had taken place, but the results of the remaining students provided a more accurate sample.) Knowing how well students had performed on each question, when taught in a variety of different styles, it was possible to compare the educational effectiveness of each approach. Since the amount of time spent adding each feature and medium to the software was known, it was then possible to identify which media had been the most efficient.

It was found that interactivity is the most vital single ingredient in CAL software. Experimental results clearly showed that learning was most likely to occur when the subjects were required to play an active role. Attractive, informative media such as

photographs and diagrams did generally help to facilitate learning, but the effect of these was comparatively minor.

The author theorises that effective computer-based education does not necessarily involve extensive use of high quality digital video and the like; rather that the means to effective computer-based learning predate the multimedia era.

Submitted in Partial Fulfilment for the Degree of
Doctor of Philosophy, September 1999
T.I.M.E. Research Institute
University of Salford
Salford, UK

A note from the Author (April, 2015)

This is a reproduction of my thesis, translated from an obsolete file format so that it could be made available electronically. Other than the inclusion of this note, I have left the content in its final form: the one with the minor alterations required by my viva panel in November 1999. The revised thesis was submitted in 2000. Obviously, taking a fresh look at the document with 15 years' perspective, I see that there are a lot of other things I might have done differently, but this is reprint – not a second edition. Other than changes forced upon me during reformatting (page numbering and some illustrations got mangled) this is the thesis in its final form. I hope you find it interesting.



Acknowledgements

Many people have contributed to this project, such as the numerous students of the Aero/Mech Department at the University of Salford, and those of the School of Engineering at Bolton Institute, who acted as test subjects. I am particularly grateful to their lecturers, for having had faith in my work.

Special thanks are due to my supervisors, Dr. Thérèse Lawlor-Wright and Prof. Nnamdi Ekere for their advice – and to the Engineering and Physical Sciences Research Council for funding the studentship.

Finally, thank you to my family, for their considerable patience and support.

Contents

Abstract	2
Acknowledgements	4
1.0 Introduction	11
1.1 A Definition of Multimedia	11
1.2 Anticipated Benefits of Multimedia	12
1.3 Impediments to the Wider Exploitation of IEM Technology	13
1.4 Thesis Structure	15
2.0 Overview Methodology	16
2.1 Research Task Aims	16
2.2 Research Process	17
2.3 Literature Survey	17
2.4 Theory Building	18
2.5 Research Scope	19
2.6 Case Methodology Adopted	19
2.7 Subjects Tested	19
2.8 Assumptions Made	20
2.9 Resources Available	21
2.10 Formative Evaluation	21
2.11 Critical Analysis	21
3.0 An Overview of Learning Theory	23
3.1 Aims of this Chapter	23
3.2 A Definition of the Learning Process	23
3.2.1 Expectancy	24
3.2.2 Attention	25
3.2.3 Coding	25
3.2.4 Memory Storage	26
3.2.5 Retrieval	28
3.2.6 Transfer	29
3.2.7 Response	30
3.2.8 Reinforcement	30
3.3 Cyclic Models of the Learning Process	30
3.4 Conditions and Practices for Effective Learning	33
3.4.1 Memorising vs. Evaluating	33
3.4.2 Information Delivery Pace	34
3.4.3 Variance in the Learners' Attention Level	35
3.4.4 Accommodating Differences Between Learners	36

3.5	Measuring the Effectiveness of Teaching and Learning	38
3.6	Conclusions	39
3.7	Research Objectives Identified	42
4.0	An Introduction to Computer Aided Learning	43
4.1	A Background to CAL	44
4.2	Anticipated Benefits of CAL	45
4.3	Quizzes in CAL	45
4.3.1	Shortcomings of Quizzes in CAL	46
4.4	CAL or IEM?	47
4.4.1	Anticipated Benefits of Multimedia in CAL	49
4.5	Adaptive CAL	50
4.6	Interactivity and IEM	54
4.7	Development Effort for IEM Software	56
4.7.1	Justifying Development Effort	57
4.7.2	Desired Qualities in an IEM Programme	59
4.7.3	Limitations on the Development of IEM Software	60
4.8	Conclusions	61
5.0	The Application of IEM Software to Promote Design for Manufacture in the Electronics Industry	63
5.1	An Overview of PCB Manufacture	64
5.1.1	Design	64
5.1.2	Board Manufacture	64
5.1.3	Placement	65
5.1.4	Soldering	65
5.1.5	Cleaning	65
5.1.6	In-Circuit Testing	66
5.1.7	Rework	66
5.1.8	Assembly	67
5.2	Case Study: Design for In-Circuit Testability	67
5.2.1	Existing Design for Testability Guidelines	69
5.3	The DICTA Information System	69
5.4	The DICTA Slideshow under Microsoft PowerPoint	70
5.5	The DICTA Information System under HiPWorks	72
5.6	The DICTA Information System under HyperCard	73
5.6.1	Dynamic Generation of Graphical Content for the DIS	75
5.6.2	DICTA Information System Flexibility	76
5.6.3	Hyperlinks between information in the DIS	76
5.7	The DICTA Information System under Multimedia ToolBook	78
5.7.1	Programmed Animation as an Alternative to Digital Video	79

5.8	Conclusions	81
6.0	An IEM Tool to Support the Education of Novice Process Planners	83
6.1	A Definition of Process Planning	83
6.1.1	Contents of the Process Plan	84
6.2	Teaching of Process Planning at the University of Salford	86
6.2.1	A Sample Exercise in Process Planning	87
6.3	An IEM Training Tool for Process Planning – The HyperCard Prototype	91
6.3.1	Results with the HyperCard Prototype	95
6.3.2	Student Responses to the Prototype	97
6.4	An IEM Training Tool for Process Planning – with Multimedia ToolBook	100
6.4.1	Colour Graphics	101
6.4.2	Changes to the User Interface	101
6.4.3	Built-in Documentation	105
6.4.4	Digital Video in CAL3P	106
6.4.5	Polygons and DCG Animation	109
6.4.6	Scale and Animations	110
6.4.7	The Built-in Quiz	111
6.5	Expanding the Scope of CAL3P	111
6.6	Limitations of CAL3P	115
6.7	Conclusions	117
7.0	Experiments Conducted	119
7.1	Assessing the Effectiveness of CAL3P Instruction	120
7.1.1	Design of CAL3P's Built-in Quiz	121
7.2	Experimental Method Employed	124
7.2.1	Knowledge Tested	124
7.3	Questions in the CAL3P Quiz	126
7.3.1	Question 1: Drilling	127
7.3.2	Question 2: Internal Boring	128
7.3.3	Question 3: Lathe Nomenclature	129
7.3.4	Question 4: Turning Problems	130
7.3.5	Question 5: Turning Between Centres	131
7.3.6	Question 6: A Simple Turning Operation	133
7.3.7	Question 7: Parting Off	134
7.3.8	Question 8: The 4-jaw Chuck	135
7.3.9	Question 9: Influences on the Process Planner	136
7.3.10	Question 10: Taper Turning	136
7.3.11	Presenting Quiz Results	137
7.4	Results Obtained from the Quiz	139

7.4.1	Results from CAL3P's 'trick questions'	144
7.5	Student Evaluation of CAL3P	144
7.5.1	Ease of Use	146
7.5.2	An Enjoyable Exercise?	147
7.5.3	Media Displayed by CAL3P	148
7.5.4	CAL3P Exercise Duration	149
7.6	Student Comments	151
7.7	Critical Analysis	153
7.7.1	IEM Development versus Improvements in Conventional Instruction	154
7.7.2	The Difficulty Level of the Quiz	155
7.7.3	Deep Learning versus Surface Learning	156
7.7.4	The Source of Students' Motivation	157
7.8	Conclusions	158
8.0	A Discussion of the Validity of IEM in Education	160
8.1	Limitations to IEM Software	160
8.1.1	Technical Constraints	160
8.1.2	Users' acceptance of IEM software	162
8.1.3	Development Effort	162
8.2	Overcoming the Limitations to IEM Development	164
8.2.1	Justifying Development Effort	165
8.2.2	Audience Size for an IEM Application	166
8.3	Development Effort for CAL3P	168
8.4	Text Elements	169
8.4.1	Good Practice in the Presentation of Text	171
8.4.2	Options Available in the use of Text	172
8.4.4	CAL3P's Text Content	174
8.5	Graphic Elements	176
8.5.1	Good Practice in the Presentation of Graphics	177
8.5.2	Options Available in the use of Graphics	177
8.5.3	Advantages and Disadvantages of Graphics	178
8.5.4	CAL3P's Graphic Content	180
8.6	Video Elements	183
8.6.1	Good Practice in the Presentation of Video	184
8.6.2	Options Available in the use of Video	184
8.6.3	Advantages and Disadvantages of Video	185
8.6.4	CAL3P's Video Content	185
8.7	Animation Elements	187
8.7.1	Good Practice in the Presentation of Animation	187
8.7.2	Options Available in the use of Animation	187

8.7.3	Advantages and Disadvantages of Animation	188
8.7.4	CAL3P's Animation Content	190
8.8	Audio Elements	192
8.8.1	Good Practice in the Presentation of Sound	193
8.8.2	Options Available in the use of Sound	193
8.8.3	Advantages and Disadvantages of Sound	194
8.8.4	CAL3P's Sound Content	195
8.9	The Value of Each Medium	195
8.9.1	Results Observed with Text	196
8.9.2	Results Observed with Graphics	199
8.9.3	Results Observed with Video	201
8.9.4	Results Observed with Animation	203
8.9.5	Results Observed with Sound	203
8.9.6	Results Observed with Interactive Learning	204
8.9.7	Software Architecture, and Learning from Tangential Information	204
8.10	Conclusions	205
8.10.1	Correlation Between Learning Theory and Results Observed	206
8.10.2	CAL3P's Most Efficient and Effective Media	208
9.0	Overall Discussion	212
9.1	Summary of Work Conducted	212
9.2	Conformance to Project Aims	213
9.3	Recommendations	215
9.3.1	Achieving Acceptance for IEM Software	216
9.3.2	Operating Within Technical Limitations	216
9.3.3	Justifying Development Effort	217
9.3.4	Further Recommendations	218
9.4	Further Work	219
9.4.1	Body-language and the Computer-based Tutor	219
9.4.2	Towards a Better Understanding of the Effectiveness of each Medium	219
9.4.3	Adaptive IEM Courses	220
9.4.4	Closer Activity Monitoring	220
9.4.5	An Investigation of how IEM Software is Used	221
9.4.6	A Future for CAL3P	221
10.0	Conclusions	223
10.1	Deliverables	223
10.2	Findings	223
10.2.1	Methodology Employed	224
10.2.2	Interactivity	224

10.2.3	Positive and Negative Aspects of the Different Technologies' Impact on IEM	225
10.3	Major Contributions to Science, Engineering and Education	226
	Glossary of Terms	229
	References	236
	Appendix 1: Multimedia Software Architecture	244
	Appendix 2: Multimedia Authoring Software Evaluation	250
	Appendix 3: Dicta Project Guidelines	256
	Appendix 4: Student Exercises and Documentation	261
	Appendix 5: A Model Answer to the Process Planning Exercise	268

1.0 Introduction

Before discussing the potential for multimedia to improve engineering education, it is necessary to establish what is meant by the term. This chapter supplies a definition, suggesting that the technology should be of value. Some factors which have significantly inhibited its widespread use are identified, to be addressed in the remainder of the thesis. Finally, the structure of the remainder of the thesis is described.

1.1 A Definition of Multimedia

Using the broadest possible definition, 'multimedia' refers to some combination of images, movies, sounds and text. Unfortunately, this leaves considerable room for interpretation. In libraries, the term is used to identify those books in the collection which have some supplementary material such as posters, slides and audio or video cassettes. As a result, one important qualification which must be added to the definition is that 'multimedia', in the context of this work, refers to a combination of digitally stored information types, ie those which are played via computer.

The technology has been applied in many fields, including advertising, entertainment, museum exhibits, corporate communication and staff training. Indeed, the term 'multimedia' has been somewhat abused in the past, used to promote software and hardware products. The basic principle, that the concurrent use of several different types of information qualifies as multimedia, is too vague and has allowed exaggerated claims to be made – one cause of disappointment among some early users.

The very flexibility of the technology of multimedia gives rise to a problem; since it can be employed for a wide variety of purposes, pieces of software cannot be meaningfully compared. Attempts to 'pin down' multimedia to a single definition are therefore understandably difficult. The definition given below is one possible interpretation:

"Interactive multimedia integrates a variety of media – including video, sound, music, photographs, drawings, computer graphics and animation – in a single electronic medium, and it provides the ability to link ideas in a non-linear format."
[Falk and Carlson, 1995]

The key section in Falk and Carlson's definition is the linking of ideas (or content) in a non-linear way. Selections made by the user/viewer influence what is seen. This interactivity is thus a key element of the definition of multimedia which will be employed

within this thesis, to differentiate from technologies (such as video) which exhibit multiple media but which can only be viewed in a linear manner.

For the purposes of this research, the application of multimedia in the field of engineering education has been considered. Hereafter, the computer-based presentation of educational material in multiple media is referred to as Interactive Educational Multimedia (IEM). Interactivity distinguishes the software from simple 'rolling' demonstrations or technologies which only allow the playback of a single, prerecorded message. Educational applications are of interest as opposed to entertainment and, finally, it will be seen that interactive educational software is not necessarily multimedia; some Computer Aided Learning (CAL) software predates multimedia, and has produced effective learning experiences.

1.2 Anticipated Benefits of Multimedia

One commonly stated benefit of multimedia (according to authors such as Luther [1994]) is that a computer which employs a mix of media will be more likely to attract a viewer's attention and thus persuade them to make use of the application. Also, Nickerson [1965] suggests 'vivid' material is more likely to be remembered. In the computer-aided learning context this suggests that employing multimedia will make for more effective training.

Multimedia allows communication in a less ambiguous fashion than might be achieved with a conventional computer programme. In this thesis, a conventional computer programme is defined as software which informs the user via a single medium, such as text displayed on the screen, or a very limited range of media – the addition of a few beeps and some diagrams does not qualify a program as multimedia. Some types of information are better represented in certain media, in software and in the world in general. Directions are best when accompanied with a map, 'wanted' posters require an image of the suspect, and so on. Multimedia technology allows people producing computer-based information to select the medium or set of media which they feel will be most effective. Hall and Robinson [1995] found video to be an ideal medium for showing the structure of the human brain to Open University biology students, so audio cassettes and a plastic model which could be dismantled were superseded with a video cassette. This was successful, but the linear nature of the 50 minute programme produced made accessing specific sections which were of interest difficult; a CD-Rom was produced which incorporated material from the video. (Distance learning should be made easier when software like this is available, but the approach can also be valuable within other

situations, such as in industry.) In the past, many staff training programmes have required that a group of employees are sent away on a short course, an expensive option which places the learners in a distracting, unfamiliar environment. With a multimedia computer, any required information can be made available within the familiar surroundings of the workplace. The specific training needs of an employee can be addressed, and they can return to work straight away.

Some benefits may not relate directly to the software itself; Haddon et al [1995] suggest that the creation of a piece of multimedia software “forces a more rigorous and thorough preparation of the material to be presented” – the investment of time spent in formally recording the knowledge of the development team may improve their ability to teach even if the multimedia project never reaches the programming stage.

Finally, an important benefit of IEM is the interactive element. The power of the computer allows selective, non-linear navigation through the information available, plus the capability to embed exercises within a piece of educational software, giving feedback on the learner’s performance. As Leonard suggests:

“Multimedia can improve significantly on other learning media such as books, audio tapes and videos, simply by adding an interactive ‘doing’ element to the proceedings.”

[Leonard, 1994]

1.3 Impediments to the Wider Exploitation of IEM Technology

Unfortunately, while it may be desirable to have IEM software, the creation of quality software of this kind can be a time consuming and expensive process. Earnshaw [1993] found that production cost for a multimedia solution was about an order of magnitude higher than that for a paper-based approach.

Effective multimedia software can be very hard to develop. Where a human instructor may hope to establish a rapport with his audience, helping him to choose how best to proceed, the developer of the computerised equivalent must anticipate the nature of the audience and allow for variation. Where a conventional textbook features only text, diagrams and photographs, its multimedia equivalent may feature all these plus animations, movies and narration when appropriate, presented in such a way that it can be navigated meaningfully by people with different learning styles and levels of ability.

With this in mind, it is not surprising that the level of development effort, when measured as a ratio against the playback time for a piece of IEM software, has been

reported as typically 300:1 and in some cases exceeding 800:1 [Marshall et al, 1994]. This level of input for the creation of IEM software has limited its exploitation. Multimedia (as defined in Section 1.1) has now been technically possible for about a decade. Robinson [1990] identified four relatively inexpensive computer platforms which could support multimedia, yet there is still a shortage of suitable titles in many fields, including engineering. In order to justify the cost of the development process involved, a piece of IEM software has to improve the quality of training which is possible, provide training where it would otherwise be impossible, or provide training of an acceptable quality on a more cost effective basis than could be achieved with conventional human instruction.

In this last case, though the development effort ratio of a piece of IEM software may be high, the calculation may prove favourable if the target audience that receives training from the software is large. However, there may be problems encountered when making IEM software available to a wider audience, such as people in other educational establishments. Trainers elsewhere may want to sequence and structure their lessons in ways which are incompatible with the software, they will consider different things to be of importance, and will demand different levels of understanding from their students. The 'home grown' multimedia package can be hard to export. If development effort for a small audience is difficult to justify, so is continued maintenance, and technological limitations can also prevent the wider exploitation of existing IEM software, as identified by Odor [1992]:

“Software rusts. Packages which worked perfectly well yesterday suddenly seize up today: systems which functioned flawlessly on the machines in your office fail catastrophically on similar machines in someone else’s.”

[Odor, 1992]

We are left with a technology which can be useful, as will be seen in the fourth chapter, but for which the initial investment of time and money can often be prohibitive. Differences in learning requirements and the hardware available for playback limit the size of the potential 'market' for a piece of IEM software. One area in which a shortage of IEM software was observed was in the education of novice engineers. It was anticipated that multimedia could be particularly valuable in this subject area, presenting realistic demonstrations of manufacturing equipment at work. Software of this kind would offer the advantages of lower operating expenses, improved clarity and increased safety. Programming and experiments have centred around creating IEM software for this field.

1.4 Thesis Structure

The next chapter describes the research methodology employed, describing the objectives of the work, its nature and scope.

Chapter 3 reviews existing learning theory; in order to discuss the effectiveness of educational software it is necessary to establish an understanding of the learning process itself. Chapter 4 details some of the commercial software which has been created to support learning activities. Some shortcomings to existing material are identified, and alternative approaches are proposed.

Chapter 5 describes an industrial problem suitable for the application of IEM, in teaching an appreciation of manufacturability issues to the designers of electronic products. Chapter 6 describes a piece of software developed to aid the teaching of process planning. Multimedia was employed to put a novice user in the role of process planner, able to learn by experimentation.

Chapter 7 details experiments which were conducted to assess the effectiveness of the various approaches to the presentation of information in the software developed, and interprets the experimental results obtained. In Chapter 8 these are contrasted with the development effort inherent in their provision. Chapter 9 discusses these results and proposes some topics for future examination, while Chapter 10 presents final conclusions.

2.0 Overview Methodology

This chapter describes the aims of the research, the resources employed, the methodology adopted and the scope of the activities which were undertaken.

2.1 Research Task Aims

During this research, it was aimed to address the following issues:

- To test the assumptions that CAL (and IEM in particular) provide instruction which is better, cheaper or more accessible than the conventional methods by which learning is facilitated. This would be achieved through the development and testing of a new piece of software.
- In the software trial process, to provide a useful educational experience for the subjects, such that the cooperation of educational establishments could be elicited.
- To identify the features of an IEM application which make the most substantial contribution to the learning process, leading to greater effectiveness in future software.
- To contrast the effectiveness of a medium with the cost inherent in its capture and presentation. (The cost of adding a feature to the software might be in financial terms, the time required, or perhaps the computer storage space required, depending on which is the most scarce resource.)
- To demonstrate alternatives to the conventional approach to multimedia software, which focusses almost exclusively on the playback of prerecorded, generic material offering only a limited learning experience. It was desired to find out if software could produce multimedia feedback 'on the fly', in response to a great variety of possible user inputs.
- To investigate the possibility of pushing interactivity as far as possible. (As work progressed, it became clear that interactivity could be as much a part of effective educational multimedia as any of the more commonly accepted constituents; text, images, video and sound.)
- To demonstrate how multimedia could be used within an interactive tool rather than in a simple one-way training process, perhaps being used regularly rather than only once or twice. In this way, IEM could be integrated into the workplace, allowing informal learning to take place.

- To investigate the applicability of learning theory (the work of educational psychologists) in a learning experience with a non-human teacher – the multimedia computer.

2.2 Research Process

Figure 2.1 shows how time was spent from the inception of the project to its completion. It had been intended to complete all experimental work within the three years for which the project was funded, but opportunities for access to test subjects were limited to specific stages in the academic calendar, so several experiments were actually conducted within what would otherwise have been the writing up period.

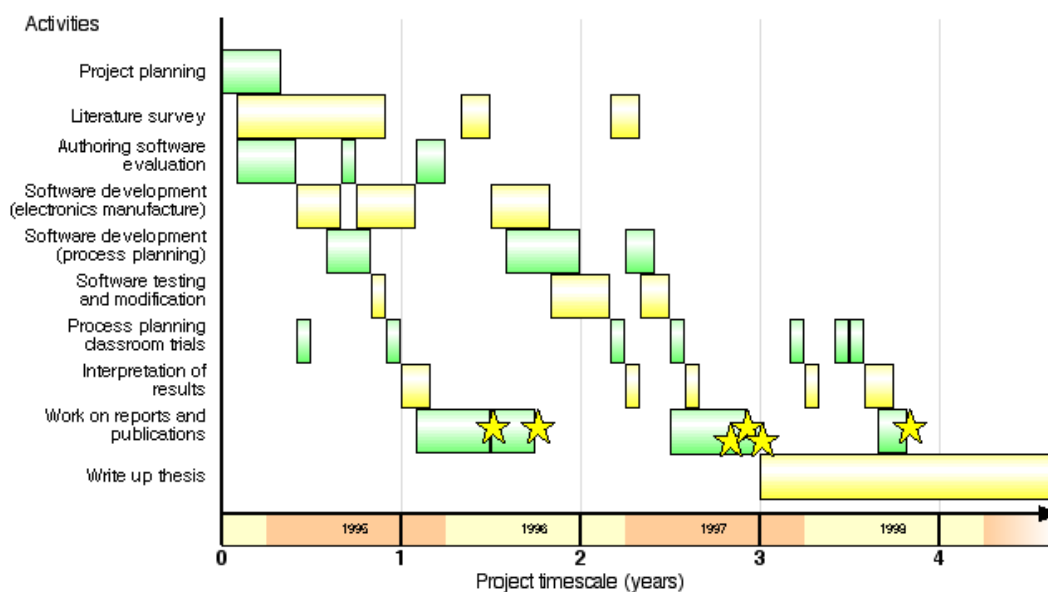


Figure 2.1: Research process timescale

2.3 Literature Survey

There were several areas requiring literature review; the first area studied was the technology of multimedia computing, in an effort to define the state of the art at the time. As the work progressed it was identified that the application of educational psychology in computer-based learning had been largely neglected, so special attention was given to this area. Later, when case studies were selected for which educational multimedia software would be developed, it was of course necessary to extend the literature review to include the engineering disciplines of process planning and electronics fabrication.

Chapter 3 presents a review of learning theory, establishing the definition of learning that is followed throughout this thesis and addressing such points as the various views of how human memory operates, the steps that can be taken to make learning more effective and how the effectiveness of these techniques might be measured. In Chapter 4 some of the key projects and publications in computer-aided learning are discussed. The early parts of Chapters 5 and 6 similarly detail design for manufacture in the electronics industry, and process planning for machining operations respectively.

2.4 Theory Building

Section 1.3 described some impediments to the exploitation of multimedia technology in education; principally that it is difficult to justify the development cost. As the literature search progressed, the author found many references to multimedia serving to attract attention, make learning more enjoyable or reduce costs (see Section 4.2), yet very few of these described any test or experiment which actually proved these claims. It was desired to conduct an experiment which would test the collective assumption that the use of multimedia will always improve learning.

It was accepted by the author that multimedia would be found to be of some value in an educational computing context; the question was – how much? Do more megabytes necessarily indicate a better learning experience? This seemed unlikely; if we allow the direction of IEM development to be technology led, we find ourselves incorporating features because we can, rather than because we have decided that we should. With advancements in digital video, for example, new developments allow higher resolution and greater frame rates, but we already have access to a technology of this kind, which we call television. Few educators would argue that watching television is the best way to learn. It was desired to investigate the possibility of finding a compromise where multimedia was informative, but required more than simple passivity on the part of the learner.

It was theorised that the amount of time (and from that, the cost) of presenting a piece of information within an IEM environment could be measured, and then compared with the educational effectiveness observed. A detailed understanding of the relative costs and merits of each constituent would represent a substantial contribution to knowledge.

2.5 Research Scope

In developing tools to meet the aims outlined in Section 2.2, two areas were considered in detail and addressed by the development of software. Firstly, in printed circuit board design it was aimed to improve product quality through the use of interactive multimedia to present manufacturing expertise at the design stage, as detailed in Chapter 5. Secondly, in the teaching of an integrated design module for university students it was aimed to prove the value of some novel approaches to multimedia delivery, as described in Chapter 6.

2.6 Case Methodology Adopted

The basis of the research was experimental in nature. It involved the creation of a piece of IEM software which students were required to use. Following this they were subjected to an attainment test which was meant to measure the amount which had been learned as a result of using the software. The test process is described in detail in Chapter 7, and results are reviewed in Chapter 8.

2.7 Subjects Tested

During this research, a large number of people acted as experimental subjects. Student involvement began in March 1995 with the author assisting in the teaching of process planning by conventional means. By December of that year, the first software prototype was introduced on an M.Sc. course. Section 7.1 details the sequence of classroom trials that were carried out. The people who took part in these trials were engineering students at undergraduate or Masters' level. As such, they were ideal subjects for the testing of a piece of IEM software for engineers. The results from these tests are presented in Section 7.4; broader analogies for the application of IEM to learners in other fields are drawn, but it is noted that the technical background of the subjects may have influenced their disposition towards computers in general and computer-based learning in particular.

Prior to these experiments, volunteers used experimental software at various stages in order to verify the user interface and content. Further, the process planning software (see Chapter 6) was demonstrated at four companies, two schools and two other universities. The DICTA Information System (see Chapter 5) was demonstrated to industrial partners at project review meetings and to electronics industry managers at

several other events such as exhibitions and open days. In many cases, comments made influenced the style or content of the software.

2.8 Assumptions Made

In order to conduct this research, it was necessary to make several assumptions, the impact of which are discussed here.

One 'experiment' which formed a part of this work concerns the creation of the IEM software employed. Since the time required to create key features of these pieces of software was measured and later compared to their educational effectiveness, the rate at which work progressed is of some importance. In effect, this measurement is based upon a sample size of one, since the author worked alone to write the software. Whether another development team would work at the same rate is not known (though Avner [1988] showed the presence of a single author is by no means uncommon for IEM software). As section 8.3 describes, development work which was measured for this purpose was carried out only after a period of familiarisation.

While every effort was made to understand the principles upon which effective use of each medium was founded (these are listed in Chapter 8), it is impossible to say that another developer might not, for example, be a better photographer or a better cameraman. When comparing the effectiveness of a variety of media, it was assumed that each was of equal quality. To be entirely even-handed with respect to each medium would be all but impossible, since the suitability of each would be affected by the subject to be taught. In no case, however, did a substantial number of students complain about a particular medium, so the assumption remains, that each medium was given an opportunity to show a measurable influence on the learning process.

It is hoped that the findings of this research can be applied to IEM on a wide range of subjects, though it must be borne in mind that all the work conducted took place within the engineering field. As a result the test subjects are more likely to be interested in and familiar with computers than those studying some other subjects. The applicability of the findings of this research to other areas, then, requires an assumption to be made; that the nature of both students and subject are sufficiently similar.

2.9 Resources Available

At the beginning of this work, resources were extremely limited. Computers of multimedia specification were still uncommon and relatively expensive. It is easy to forget that in 1994 when the project began, equipment such as a CD-Rom drive or even a sound card was by no means standard equipment. Processors, memory and hard disk space were similarly limited.

For a time, there was no multimedia-capable computer available to the author. Involvement with the DICTA project (see Chapter 5) solved this problem, allowing a computer and authoring software to be purchased. The lack of multimedia capability is not merely a problem at the authoring stage, however; if a piece of IEM software is to be employed in a classroom situation then a large number of multimedia computers is required, and keeping these up to date is correspondingly more expensive. It was found that the hardware in the computer suites of educational establishments tended to lag behind by a generation or two; making educational software which was still useful on these lower specification machines became a theme of this research as a result.

2.10 Formative Evaluation

Bearing in mind the assumptions made, as described in section 2.8, the approach seemed to be sound. It would tackle head-on the assumption that technology could lead to better instructional techniques despite the fact that human learning theory was all too often disregarded. At this stage in the research it remained to be seen whether that science would prove relevant in instruction of this kind, and the author had yet to demonstrate that an effective piece of IEM software would result at all.

2.11 Critical Analysis

It might be argued that more accurate measurements of educational effectiveness could be made by requiring learners to study an entirely synthetic subject; by creating a fictional topic of which none of the learners could have any prior knowledge at all, any knowledge which remained afterwards would indicate that the material was presented in an effective manner. However, by offering something which was beneficial to the learners, their motivation was increased and they were more likely to concentrate on both the learning process and the test which followed. This also had the advantage of ensuring the

cooperation of course tutors, without which it would have been much harder to find a sufficiently large group of test subjects.

3.0 An Overview of Learning Theory

Before exploring the possibilities offered by CAL in general, and IEM in particular, it is necessary to have an understanding of the process to which the computer is being applied. Indeed, what does 'learning' mean? This chapter surveys theories which have been proposed to explain how learning takes place. With these in mind, key points for the development of computer-based information resources such as IEM software are identified.

3.1 Aims of this Chapter

In discussing the process by which humans learn it is first necessary to establish what is meant by learning. Basically, learning means gaining knowledge through study or experience. It might also refer to furthering one's understanding of knowledge already held – gaining comprehension. This process of acquisition of facts and mastery of skills has many different names; finding out, memorising, studying, understanding and others. All refer to the process of forming meaningful associations between pieces of information as they are stored in the brain. Learning is a very complex endeavour which is still only partially understood, but specific knowledge in this field can be accumulated by scientific method. Research in educational psychology has produced theories which aim to explain how learning takes place, and thereby improve the process. This chapter reviews learning theory in an attempt to provide answers to the following questions:

- What is the nature of the learning process?
- What are the conditions required for effective learning?
- How can testing be conducted to measure learning that has taken place?
- How can knowledge of learning theory be applied to improve the effectiveness of instruction?

3.2 A Definition of the Learning Process

Gagné defined learning thus:

“... a process which enables ... organisms to modify their behaviour fairly rapidly in a more or less permanent way, so that the same modification does not have to occur again and again in each new situation.”

[Gagné, 1975]

Similarly, Baddeley [1990] describes learning as the modification, by experience, of behaviour (though he notes that the altered behaviour might not be demonstrated at the time of the act of learning).

There are two opposing viewpoints on the influences to the learning process; empiricism and rationalism [Hilgard and Bower, 1975]. Empiricism implies that experience is the only valid source of knowledge, while rationalism suggests that reasoning is the prime source of knowledge. Though both principles have had their adherents in the past, it would appear that the most realistic position lies somewhere between the two. As Kant [1781] stated: “Although all our knowledge begins with experience, it by no means follows that it all originates from experience.” This has been confirmed by studies of language development in children, showing that not all learning is experiential, since most children eventually pick up good language skills despite the poor quality of ‘teaching’ they receive from parents, who tend not to correct mistakes and often indulge in ‘baby talk’ [Hilgard and Bower, 1975]. Clearly, the child exercises an ongoing process of evaluation as new words and phrases are heard. Thus, learning is believed to involve receiving inputs, and ‘filing away’ the understanding that is derived from them. Gagné [1975] offered a learning model made up of eight phases, as shown in Figure 3.1. Each phase is addressed in turn in the subsections that follow.

Time ↓	Phase name	PROCESS
	Motivation	EXPECTANCY
	Apprehending	ATTENTION
	Acquisition	CODING, STORAGE ENTRY
	Retention	MEMORY STORAGE
	Recall	RETRIEVAL
	Generalisation	TRANSFER
	Performance	RESPONDING
	Feedback	REINFORCEMENT

Figure 3.1: The phases of an act of learning [Gagné, 1975]

3.2.1 Expectancy

Naturally, the learning process requires a motivated individual, so before actual teaching begins it is necessary to have incentives established. Thus, the learner is working towards an achievable goal which is held to be of some value. When motivation is absent in the learner, they must be brought to a point of expectancy, being given an appreciation of the reward which will result from successful learning. Activating motivation requires that the knowledge which will be gained leads towards some desirable objective. The

reward which is established need not be tangible; there is a natural human tendency which urges mastery of new challenges. In motivating a person to learn, it will be necessary to relate to their current situation. Thus, motivation requires relevance which may vary between individuals at different times and in different circumstances.

3.2.2 Attention

Having established the outcomes that may be expected from study, in the apprehending phase, the learner must give attention to the stimuli which are presented as part of the teaching process. Different senses may be stimulated depending on the media employed in the learning process and it will be necessary for the learner to disregard distracting influences [Trabasso and Bower, 1975].

Bradshaw [1997] draws a distinction between getting a user's attention and holding their interest. This may be a more accurate term for what is needed in education, where problems or case studies are presented, requiring active involvement on the part of the learner.

3.2.3 Coding

With attention being paid to the correct stimuli, acquisition can take place. This is the stage in which information enters into the short-term memory by coding – translating it on receipt into a form which is suited to the learner. The stimuli which are presented at this point can seldom be accurately recalled in themselves, even though an 'act of learning' takes place. A learner who is asked to repeat what he or she has just been told might be able to supply the gist of the message, but is unlikely to reproduce it word for word. Thus, coding is the change of the information provided into a form understood by the recipient.

Hunter [1957] carried out an experiment which illustrates the coding process particularly well; subjects were given one minute to memorise an image of nine numbers arranged in a grid (Figure 3.2), which they were requested to reproduce an hour later:

Naturally, some people succeeded and some failed to reproduce the image accurately. Those who had succeeded were asked how they had managed to remember the number sequence. Methods varied. '492' was the last three digits of the year in which Columbus made his historic voyage; the even numbers were all in the corners; some noted that the numbers formed a 'magic square' with any line of three numbers adding up to fifteen... The methods by which coding had taken place were diverse, but nobody – even among those who had failed – had tried to learn in a passive manner such as by simply memorising the array of numbers without looking for underlying rules. Presented with a

sample of information which must be learned, the subject tries to encode; to impose some form of order on the data [Hunter, 1957].

4	9	2
3	5	7
8	1	6

Figure 3.2: Figure used in a memory exercise [Hunter, 1957]

3.2.4 Memory Storage

Initial acquisition places information in short-term memory (STM) which is severely limited in capacity, with information surviving for only a few seconds before being displaced as more arrives. Ergonomics research by the USAF suggests that most people are able to store 5 ± 2 pieces of information in STM, for 15 to 30 seconds [Fisher, 1994]. Others, such as Atkinson et al [1993], suggest that STM may be able to store as many as 7 ± 2 pieces of information. Murdock [1965] and Peterson and Peterson [1959] both show recall performance of a three consonant sequence typically falling to less than 20% after 18 seconds, clearly demonstrating the transitory nature of short-term memory.

Learning, as defined in Section 3.2, requires that behaviour is changed “in a more or less permanent way”. The initial acquisition of short-term memories cannot account for this. Thus, the memory storage process is identified. This involves a further translation to allow the transfer of information into long-term memory. Various coding techniques are used to improve performance such as finding patterns within the information or relating it to other concepts which are already understood; Ausubel [1968] states that for individual facts to be learned most effectively they must be ‘subsumed’ in a larger meaningful context, which is to say related to other facts which are already known. Artificial techniques such as the creation of mnemonics can also be employed.

Effective coding for long-term storage can be promoted or inhibited by the media used to stimulate the learner. Studies have shown that memory of visual stimuli is particularly good. In one notable experiment by Haber [1970], subjects were shown a series of 2560 photographic slides, each for 10 seconds, over a period of several days. Testing then involved showing each subject 2560 pairs of slides, one which they had been shown previously and one which was similar but unseen. The subjects were required to identify the previously seen slides. Accuracy of recognition was seen to be at 85–95% on average. Haber carried out further experiments, reducing the time for which a slide was

seen to just one second, and also showing them as mirror images. Results were identical, suggesting that the human assimilation of information from pictures is very fast, and can involve some ‘juggling’ of the image without problems. Nickerson [1965] used similar methods and found an average accuracy, on testing immediately after presentation, of 98%. Expanding the number of pictures up to 10,000 images resulted in no reduction in accuracy. There was no indication of the memory capability of the human brain being exceeded.

Levin and Kaplan [1972] demonstrated that pictures suggesting visual images could be a highly effective means of coding, allowing a complex piece of information to be stored in memory as a single, simple concept. Figure 3.3 shows how images of similar complexity (the same number of connected straight lines and curves) may or may not be easy to remember, depending on whether they can be encoded. The abstract collection of lines (a) is unlikely to be recalled a few minutes later, but the second image (b) promotes coding and hence recall because it can be conveyed as a very simple concept. It does not matter if the viewer thinks of the image as a tent, a rocket, a fat pencil stub or something else. They have been able to perform a coding process and are therefore more likely to be able to reproduce the image on request.

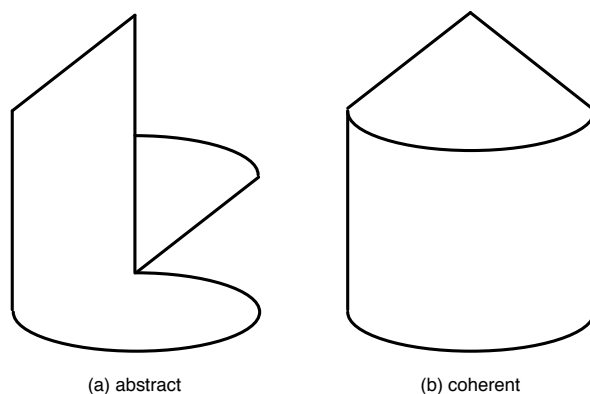


Figure 3.3: Example of how images of similar complexity may be (a) harder or (b) easier to encode effectively.

Carmichael et al [1932] demonstrated how it was possible to influence learners’ coding process using language. Test subjects’ perception of visual sources was biased by giving the subjects different suggested names for each of a series of partially ambiguous figures such as that shown in Figure 3.4. Some subjects were told the shape represented ‘eyeglasses’, while others were told the shape was a ‘dumbbell’. Subsequent sketches by the test subjects commonly showed the figure changed to conform to the coding which had been supplied.

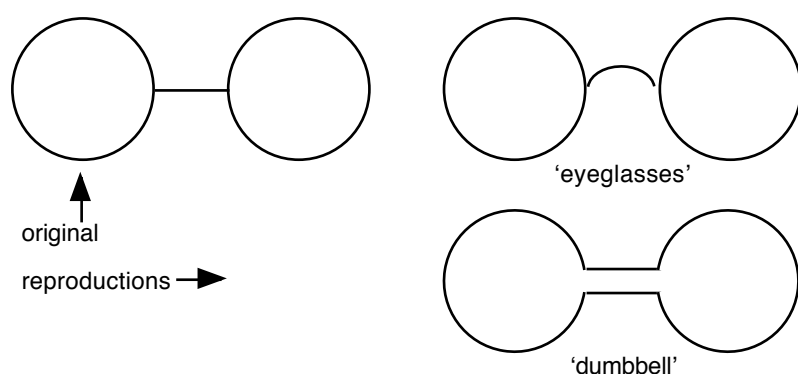


Figure 3.4: Changes in figure reproduction brought about by suggested encoding [Carmichael et al, 1932]

Clearly, it is possible to influence how an image is stored in memory and later recalled through the use of a secondary medium, something which will be of particular significance when multimedia is employed in the learning process since the definition of multimedia requires that several stimuli can be delivered simultaneously.

3.2.5 Retrieval

The retrieval phase begins when an attempt is made to make use of the information which has been stored. If retrieval cannot be achieved, a skill cannot be demonstrated. Cues for retrieval can come from external stimuli ('jogging the memory'), or from other associations within the brain. A failure to retrieve information is often called forgetting, though the mental processes which lead to forgetting are less simple than might be imagined. It is important to distinguish between information which has actually been forgotten and that which was not successfully learned in the first place. Much which is written off as forgotten may not have been given sufficient attention when the stimuli were provided, so adequate learning did not take place.

Baddeley [1990] identifies that the theories which have been proposed to explain forgetting fall into one of three groups; those of interference, trace decay and fragmentation. The interference theory suggests that memories are overlaid by more recent ones and so become hard to recall. The trace decay theory implies that memories erode naturally over time, changing rather than disappearing completely. The fragmentation theory (Bower [1967] called it the multicomponent theory) surmises that memories 'crumble' into separate parts and are lost piecemeal as the associations between parts weaken.

Since memory is an internal function it is difficult to test directly, but experiments can be carried out and data obtained. Ebbinghaus [1913] observed a logarithmic decline in ability to recall a list of syllables over time, but subsequent results suggest that the rate at which recall performance deteriorates depends on the nature of the skill or information being tested. Linton [1975] demonstrated an essentially linear forgetting function. Testing of continuous motor skills by Fleishman and Parker [1962] shows virtually no decay over time (hence the well-known saying, "It's like riding a bicycle"). Foreign language skills were seen to decay quickly at first, but then steady after two years with the remainder subject to little decay [Bahrick, 1984]. McKenna and Glendon [1985] observed a time-based performance drop among first aid volunteers trained to carry out cardiopulmonary resuscitation; after one year the skills learned had decayed to the point where the survival rate for a resuscitable patient had fallen from 100% to 15%. Scores for performance showed a sharp decline during the first six months, followed by a steady decline to almost complete uselessness after three years.

Understanding the nature of the forgetting function for a skill could supply valuable information to the designer of IEM software, such as how frequently the software should be reused. Rehearsal, the occasional use of a fact or skill stored in memory, strengthens the associations within the brain and increases the likelihood of successful future recall. It may be valuable to use computer simulation in the rehearsal of a safety procedure which it is vital that the learner understands, but where the catastrophic problem occurs very seldom in real life.

3.2.6 Transfer

Transfer is necessary because the recall of information will inevitably be required in circumstances which are different from those in which the original learning event took place. Completion of a test which requires transfer confirms that a principle has been learnt, rather than that an earlier answer was simply memorised. In order to recall knowledge so it can be applied in the new situation it will be necessary to generalise. Transfer allows retrieval of useful information where the stimuli jogging the memory are an imperfect match, allowing judgment as to whether the knowledge is appropriate. By attempting to apply knowledge in a new situation, further learning can take place.

Effective transfer may be impeded by the environment in which recall is attempted. Estes [1972] stated that it is easier to retrieve a particular fact in the same context. This was demonstrated by Godden and Baddeley [1975], who found that recall performance dropped by 40% when subjects were tested in an environment which was different to the one in which they had learned. A substantial part of the value of multimedia in computer-

aided learning may lie in its ability to present a learning situation which is similar to the environment where the new skill must be employed.

3.2.7 Response

Having determined that some knowledge is relevant, it is brought to bear by the process of response. Response might involve consciously making decisions based on learned information (for example, demonstrating a newly acquired skill by solving simultaneous equations), or it might be automatic (such as when hitting a ball with a bat). In either case, it is at this stage that learning is demonstrated, and only when a skill has been demonstrated can it be considered to have been learned. Even to the learner who may be highly confident, a test of actual performance is the best way to be assured that they can rely on the newly acquired skill in future. Satisfaction comes from success at this stage.

3.2.8 Reinforcement

Although the learner may be satisfied with their performance, it is preferable that their responses are assessed independently in case there has been some error in the learner's understanding. In this feedback phase the successful learner receives the expected reward, as established in the motivation phase. This acts to condition the learner towards repeating the process and is consequently called reinforcement. If performance has been unsatisfactory the reward does not materialise, or is withheld. Punishment for exhibiting undesired behaviour is called negative reinforcement, and acts to condition the learner against repetition.

3.3 Cyclic Models of the Learning Process

Kolb [1974] presented a model in which learning was seen as an ongoing, iterative process. Kolb's Learning Cycle is shown in Figure 3.5. Experiences are reflected upon (the empirical data being subjected to rationalisation), and from these reflections an understanding is reached, to be tested in future experiments which will in turn generate new experiences. Thus, Kolb presents a view of learning as continual testing during life experience, with modification of behaviour as a result of observations made.

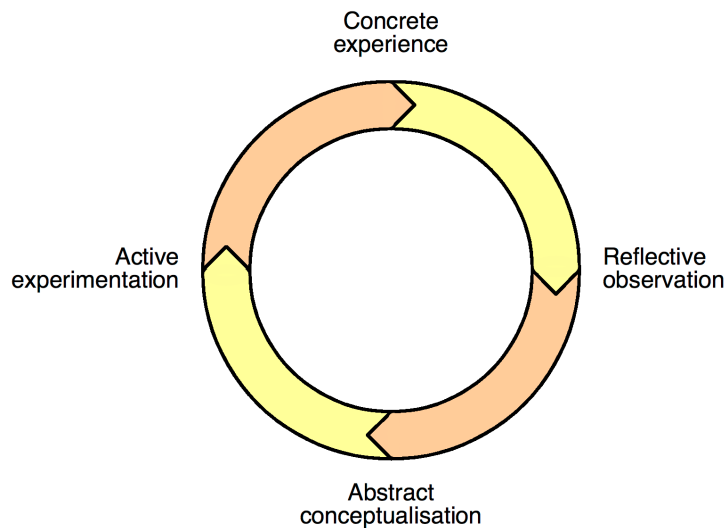


Figure 3.5: The Kolb learning cycle [Kolb, 1974]

For the development of a CAL tool, Crampes [1991] produced a modified Learning Cycle (Figure 3.6). This model differs from Kolb's Learning Cycle in the terminology employed and the distinctions between phases, though both present complete cycles. Crampes' 'discovery' phase encompasses Kolb's 'reflective observation' and 'abstract conceptualisation' phases. Similarly, Handy [1989] presented the 'Wheel of Learning', with the four phases question, theory, reflection and test. Handy suggested that it is possible to become overly reliant on one of the phases. For example, one might think too much and seldom act, or experiment compulsively but pay little attention to the results. Any such failure to work through a full cycle inhibits effective learning.

Yolles [1997] suggests that the Crampes Learning Cycle is valid in a dynamic learning situation such as simulation; a student works to build up an understanding of a system by making inputs and observing results. Their hypothesis is evaluated and reevaluated until the simulation is understood, at which point the learning objective has been reached and the cycle ends.

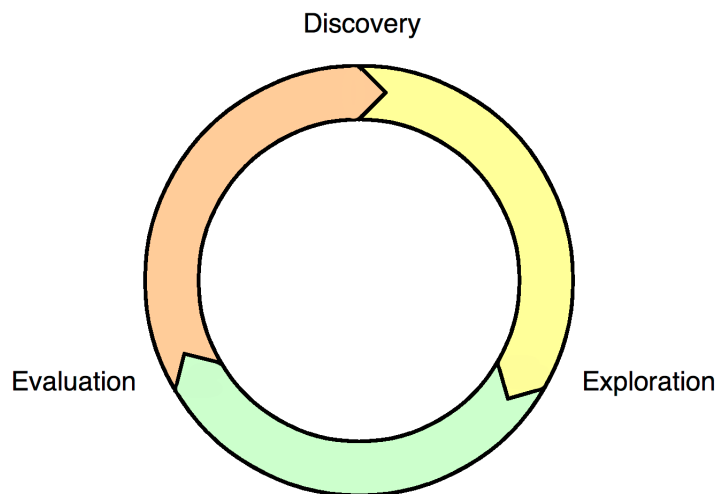


Figure 3.6: The Crampes learning cycle [Crampes, 1991]

The models of learning as a cycle are not incompatible with the Gagné's eight phase model, though some of the boundaries do not coincide, or are hard to define exactly. The learning experience begins with expectancy being established, though reinforcement subsequently motivates the learner towards further improvement. The author proposes Figure 3.7, showing how the Kolb and Gagné models might correlate.

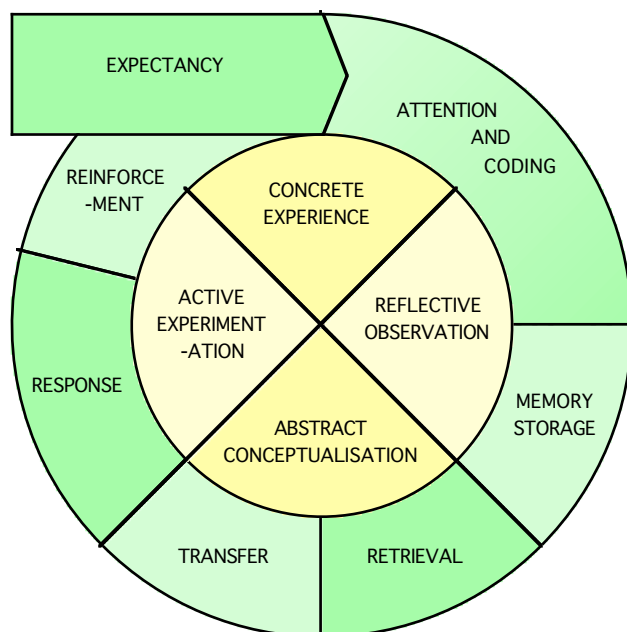


Figure 3.7: Possible correlation between the Kolb and Gagné learning models

3.4 Conditions and Practices for Effective Learning

If the learning process is sufficiently understood and modelled, it should be possible to propose a set of conditions under which instruction should be carried out. Derville [1966] suggests a difference between ‘giving lessons’ and ‘teaching’; Derville’s definition of giving lessons involves talking, writing and drawing on a blackboard, demonstrating, asking questions and setting exercises. By Derville’s definition of teaching, in addition to these information provision activities, the learners must be stimulated to play an active role themselves. What Derville appears to be demanding could be described as a higher degree of interactivity in the learning process. The presentation of information in a one-way flow is insufficient. For this reason, practical exercises are valuable. The ‘information-processing’ theories of how the brain works (such as the model shown in Figure 3.8) include a feedback stage where the subject acts on stimuli received to influence their environment. This model fits well the cyclic models presented in the previous section, because it allows for iteration within the learning process through observation and experimentation.

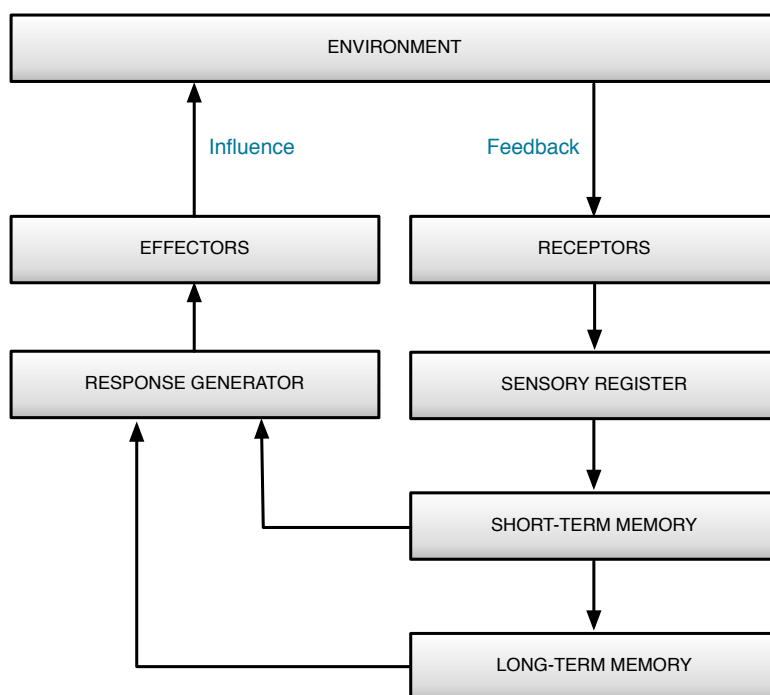


Figure 3.8: Influence and feedback in Gagné's [1975] 'Information Processing' theory of learning and memory

3.4.1 Memorising vs. Evaluating

Seale [1997] reports that a problem has been identified in some science degree curricula, where even successful students often understand much less than they have

learnt. Learners can memorise information without evaluating it, and often believe it necessary to do so as they work to meet the short-term objectives of their course. This may produce temporary memorising of facts rather than deep learning. Effectively such students are not attempting to learn, but to meet a required performance level which is being measured by a somewhat inappropriate process.

Information memorised in this way is believed to be available only in the short-term because the associations which have been formed with the new material are weak. In simply memorising information, the task of evaluating why a certain answer is right can be neglected, and the result is a qualified individual who has demonstrated some intellectual ability but has not developed the underlying practices which are necessary for a professional [Seale, 1997]. Existing IEM software, featuring the conventional approach of informative sections followed by quizzes, may similarly demand only the short-term memorisation of content rather than reflective learning.

3.4.2 Information Delivery Pace

Seale's observation that workload can cause problems which inhibit reflection in learning introduces another variable; the pace at which information is delivered. Honigsbaum [1992] suggested that level of skill could be plotted against level of challenge during a learning process, defining a region within which learning is most enjoyable and therefore most likely to take place, which he called the Flow Channel (Figure 3.9). When skill exceeds the level of challenge, boredom prevents effective learning. When the reverse is true, anxiety or frustration prevent effective learning.

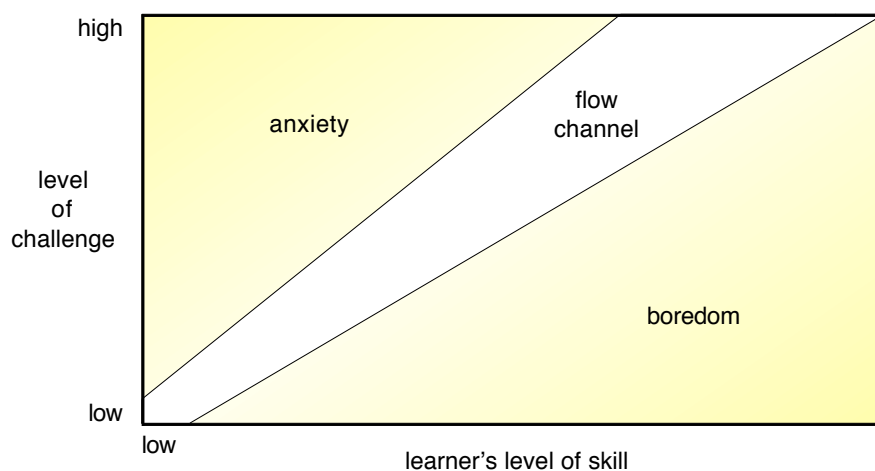


Figure 3.9: The 'flow channel' where effective learning can occur [Honigsbaum, 1992]

Adcock [1990] identifies determining the duration of a training period as one of the key inputs of the teacher, striking a balance between pressing on to further progress when something is understood and avoiding fatigue. Adcock [1990] suggests that changes to the distribution of learning periods may improve learning achieved by several hundred percent.

3.4.3 Variance in the Learners' Attention Level

It is not merely pace which influences the amount learned, but also the timing of individual acts of learning within the information delivered. Buzan [1993] suggests that the human brain will best recall items which fall into the following categories:

- Items from the beginning of the learning period (the primacy effect)
- Items from the end of the learning period (the recency effect)
- Any items associated with things or patterns already stored, or linked to other aspects of what is being learned
- Any items which are emphasised as being in some way outstanding or unique
- Any items which appeal particularly strongly to any of the five senses
- Those items which are of particular interest to the learner

In Figure 3.10 it can be seen how Buzan [1993] hypothesised that a learner's attention span might vary over a 2-hour period. The primacy effect can be seen clearly, with a good proportion of the information which is delivered at the initial stage being recalled. At the end of the session, the information most recently supplied is also more likely to be available for recall. In between, there is a slump, though some peaks exist, showing topics which are of interest to the learner for some reason.

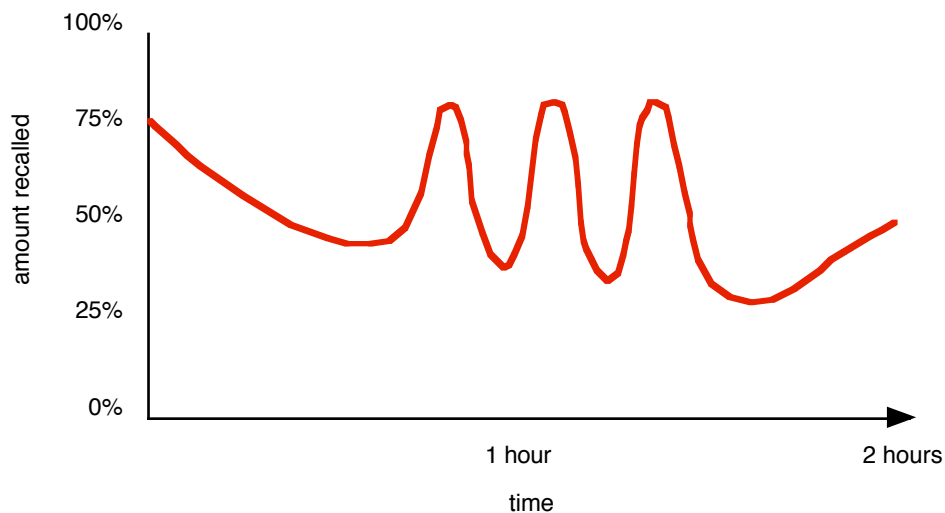


Figure 3.10: Varying level of attention over a 2 hour period, determined by the amount available for subsequent recall [Buzan, 1974 and 1993]

The first few minutes spent on a subject are clearly valuable. One technique to create multiple primacy effects is to have frequent breaks in study, often accompanied by interactive practical application. Also, it would clearly be desirable to have as many of the ‘peaks’ of concentration as possible occurring during the session. Essentially these will occur when either (a) something remarkable is observed, (b) something within personal interest is encountered, or (c) some information which is related to something already known is encountered.

3.4.4 Accommodating Differences Between Learners

Naturally, learners all have different strengths, weaknesses and interests. In addition to Buzan’s model of the level of attention paid by a learner over time, we could consider a classic interpretation of the ‘learning curve’, representing a change in competency over time. Atkinson et al [1993] describe a learning curve as “A graph plotting the course of learning in which the vertical axis plots a measure of proficiency while the horizontal axis represents some measure of practice.” Figure 3.11 (by the author) shows a generic learning curve, illustrating how a learner’s performance might appear if he or she were tested at various points during a training course. The exact shape of the curve will be hard to predict, but there should be a trend towards increased level of skill over time. This is not at odds with Buzan’s [1993] model of the level of attention paid, since attention is only one part of the learning process and does not guarantee that an act of learning will occur.

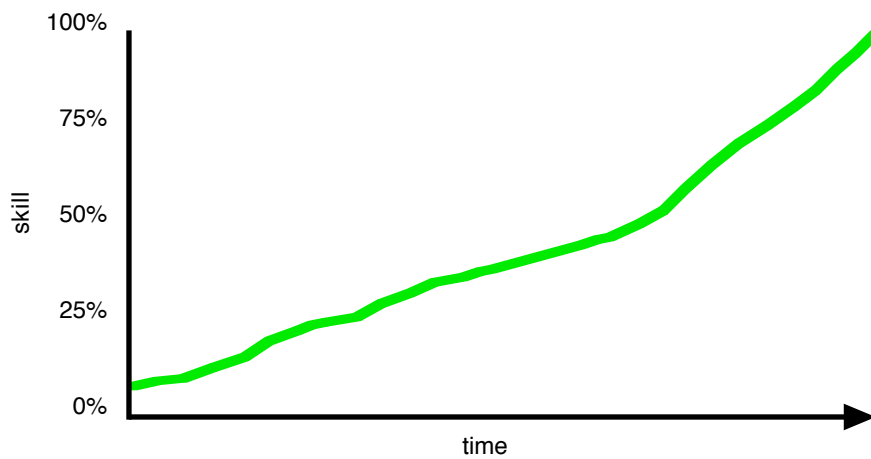


Figure 3.11: Learning curve concept

Derville [1966] identifies that a plateau of learning may sometimes be reached, where a student fails to advance in performance for a while, perhaps because they do not understand some new principle, or because they are waiting for some input. Motivation may suffer at this point, but equally a lack of motivation may be the cause of a plateau. While some students may learn faster than others, and some may begin with a higher level of skill, it is hoped that there will be some improvement in skill over time. Figure 3.12 (by the author) shows simulated learning profiles for several different people.

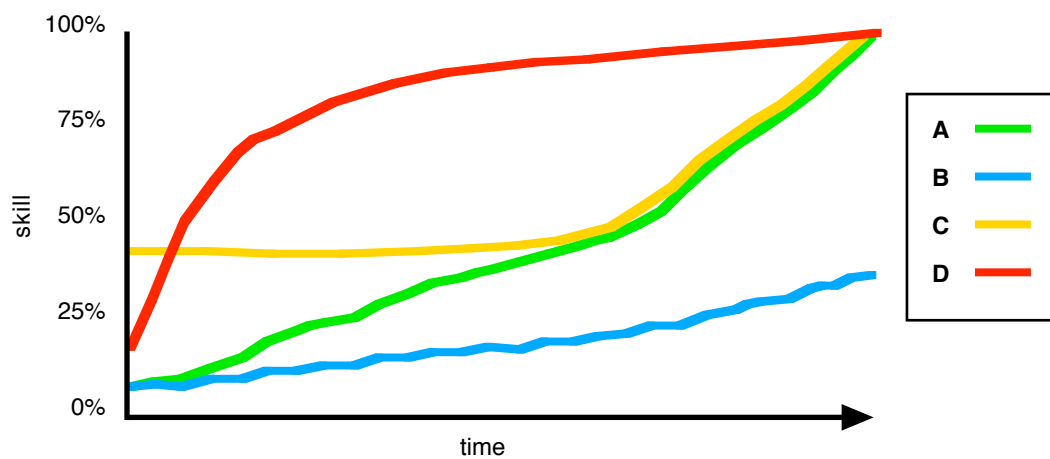


Figure 3.12: Learning curves for several hypothetical students

If 'A' is the normal learning curve for a typical student, 'B' represents one who is diligently struggling. This student's level of skill increases over time, but more slowly than the norm. 'C' is a student with some prior knowledge of the subject, who consequently doesn't get much value from the information which is provided, until new

material is reached. This student can then learn normally. 'D' is a student who surges ahead through hard work or aptitude, but reaches a point where further learning is difficult due to lack of support, and enters a stage where a diminishing return for effort is seen, possibly resulting in a loss of motivation.

Wheldall [1983] described a classroom situation in which motivation was found to be improved with additional workload; observation of a group of children doing mathematics problems had shown that only 55% of their time was typically 'on task'. One of the problems was that those who had reached the end of the work set would disturb the others. When additional problems were set for the quicker students, mean student time spent 'on task' was observed to have improved to 70%.

3.5 Measuring the Effectiveness of Teaching and Learning

In using learning theory to propose changes to a teaching process it is necessary to assess the effectiveness of the alternative instruction which is given. This will be particularly important where the possibility of introducing IEM software is raised since it represents a significant departure from conventional instruction.

Unfortunately, a subject's level of learning cannot be measured. Only performance can be measured, and the level of learning inferred from the results obtained [Hilgard and Bower, 1975]. As section 3.1 has shown, learning is subject to a number of variables over many different stages; where many opportunities for learning exist, we select the ones which we are motivated towards, and direct some portion of our attention to the stimuli offered. An imperfect translation of these simplifies the exact information into concepts. Some of these concepts are particularly memorable, or easily related to information which is already known, so these are successfully stored for later recall. Only if the correct associations are made will the information be recalled at the right time, and not all learners will be able to make the right generalisations to apply their knowledge in the right way. Having acted, only a subset of learners will receive feedback which enables them to modify future behaviour as necessary. Consequently, learning is a slow process with only a fraction of what is studied being successfully assimilated at the first attempt.

In measuring the effectiveness of a learning process it will be necessary to carry out attainment testing [Derville, 1966]. Tests should be standardised, which is to say that they should be supported by documentation ensuring that the test is administered, timed and scored in the same way each time it is employed. Model answers should be supplied to ensure consistency, and anticipated 'normal' performance in the test should be described; average scores can be found and any problems identified by carrying out testing on a

sample in advance. Having measured a person's performance in a test, the question arises as to whether the subject was taught effectively, or they were particularly bright. Thus, some testing then requires that the IQ of the subject is measured.

Derville's studies concerned school children; a further complication with adult learners such as those on degree courses is that their ages and backgrounds are likely to vary widely. Section 3.4.4 described how an initial 'head start' in a subject can require that a learner waits for the majority of the class to catch up, giving rise to problems in motivation. Of course, this is true for any form of instruction. For CAL, addressing this potential problem may represent a niche where the strengths of computer-based instruction can be brought to bear; because each computer instructs only a single learner (or a small group at most), the pace of information delivery can suit the learner.

3.6 Conclusions

In presenting established theories of how the learning process takes place, this chapter has identified a number of good practices for conventional instruction which may also be applicable in computer-based instruction. The guidelines that follow represent a summary of these findings, and were adopted for the programming work undertaken in the course of this research.

Section 3.2 described how learning has both experiential and reflective elements. Ideally, a learning environment should allow both of these to take place. Since, as Chapter 1 identified, the creation of multimedia software can be very time-consuming, it is tempting to think simply in terms of the facts which are to be presented. In this case information is gathered – in itself a significant task – and strung together in some form which a user can browse through. This approach may get the software finished, but does not satisfy Seale's [1997] requirement for reflection/evaluation, nor Yolles' [1997] demands for iterative experimentation. Doing rather than watching should result in more effective learning since attention will be increased and coding will be more likely to take place, and since the human learning process is essentially cyclic (e.g. Kolb, [1974]), it may be useful for IEM software to accompany the user through several iterations of a complex learning process.

Motivation was identified as necessary for effective learning, establishing what a learner can expect to gain when they invest their time. This is particularly important for computer-based learning, since this form of study may be unsupervised. It should also be comparatively economic in terms of development effort and computer system resources to

establish a 'deal' with the user, such as, "Once you have completed this exercise you will have learned how to..."

A key aspect of an act of learning, closely related to motivation, is reinforcement. Upon completion of a learning task, a reward is given (or withheld). Although IEM software is unable to provide any tangible reward, it could be programmed to check answers made in response to a built-in quiz and give results with individualised feedback very quickly. A typical 'reward' in computer-based learning is that the user is allowed to proceed when they have demonstrated that they have learned the material in the current section.

The pace of an exercise and level of challenge which it presents must be set appropriately to the level of competence of the audience. A computer-based learning situation offers an advantage in that users can proceed at a faster or slower pace if they wish. Wheldall's [1983] observation that overall class performance improved when additional exercises were available for the faster students may be applicable – though Travis and Jones [1997] observed behaviour where struggling students skipped through pages of information so they could appear to keep up with their classmates.

Learners can be aided or inhibited by the delivery style of information, which may promote or hinder effective coding. Particularly encouraging for the developer of IEM software is the idea that people learn best when presented with vivid experiences, and particularly images. The multimedia computer is one of the best ways to present these vivid experiences within the classroom.

In addition to learning, its inverse has also been touched upon within this chapter. Forgetting is natural and largely inevitable, though it can be postponed by occasional revision or ongoing use. It occurs at different rates for different kinds of information, and it may be that for some subjects IEM software can be employed to provide the occasional rehearsal which helps to stave off forgetting. Buzan [1974] presented a pattern of review which maintained knowledge even though the interval between rehearsals increased sharply.

Many of the factors which learning theorists have identified as having an influence on the learning process may be regarded simply as 'common sense' by teachers. However, many pieces of software do not have the qualities which learning theory suggests. In this chapter it has been seen that communication in learning should be bidirectional, yet Novak and Desharnais [1998], identifying interactivity as the essence of the pupil/teacher relationship, found current software implementations lacking. Knowing what is best and actually achieving it within software are entirely different. Turnbull [1974] reported:

“Claims of interactivity and freedom are often followed by largely linear software in which the only difference is the medium.”

This problem has not disappeared despite a massive increase in computers’ capabilities. Fisher [1994] protested:

“Too many people put a linear book on–line, give it some electronic bookmarks, and call it hypertext; worse yet, they add a few scanned–in photographs and a soundtrack and call it multimedia.”

A scarcity of academic papers addressing the application of learning theory to CAL software suggests that learning theory has been largely disregarded. Hudson [1984] warned:

“Few CAL program writers appear to have paid much attention to the important features of the human learning process now being defined by scientific research in several fields.”

Similarly, Bradshaw [1996] suggested:

“There are problems in the design of CBL programs which arise from the unsystematic, intuitive practices by which the tutor/author decides what learning activities the students are to encounter. In short, the hit-and-miss approaches of traditional teaching, although perfectly adequate for the real classroom, are inadequate for CBL design.”

As the chapter that follows shows, there are other factors influencing the developer of CAL or IEM software, such as the technical limitations of the delivery system. The creation of educational software requires mastery of computer programming and the subject to be taught – small wonder that few of those involved also have knowledge of educational psychology. Perhaps too many budding CAL or IEM creators have in mind a simple view of the CAL concept, which inherits much from a recent past when software was necessarily constrained by what was technically feasible, rather than what was best for the learner.

3.7 Research Objectives Identified

Following the discovery of a wealth of learning theory, much of which appears to have gone unnoticed by the majority of IEM developers, a number of interesting opportunities for research presented themselves. At the heart of these were the educational psychologists' models of the learning process; could they be applied in an environment where the human teacher is several stages removed from the learner? The following objectives were to be pursued in the remainder of the research:

- To investigate the nature of expectancy and reward in a computer-based learning experience; can a reward of some kind be offered, and if so would it motivate the learner? It may be that a simple, prerecorded response from the machine is inadequate for this purpose.
- To investigate how the coding process (manipulation in short-term memory) is affected when learning takes place within an IEM environment.
- To test, within an IEM context, the findings of Nickerson [1965] and Haber [1970], which imply that images are tremendously more effective than other media (see Section 3.2.4).
- To present an approach by which attainment testing can be conducted to measure the amount of learning that has taken place while a piece of IEM software was used.
- To examine the validity of educational theorists' models of the learning process. Would it be preferable, for example, to offer a cyclic learning process such as that proposed by Kolb [1974] within an IEM environment?

In the chapter that follows, a number of pieces of CAL and IEM software are examined with a view to identifying whether successful, effective programs conform to the guidelines which learning theory offers.

4.0 An Introduction to Computer Aided Learning

A computer may be employed to assist learning in a wide variety of ways. No simple definition could describe the capabilities a CAL application might possess. The Oxford Reference Dictionary of Computing attempts a definition of CAL as:

“Any use of computers to aid or support the education or training of people. CAL can test attainment at any point, provide faster or slower routes through the material for people of different aptitudes, and can maintain a progress record for the instructor.”

[Illingworth et al, 1990]

This is correct only in that a CAL application may feature attainment tests and store the marks obtained for the tutor, and may allow the pace of learning to be varied, etc. Such capabilities are by no means prerequisites for software of this kind. The Collins Dictionary of Information Technology [Collins, 1996] simply describes CAL as “use of a computer to assist pupils in learning a subject.” The only additional distinction which might be added is that the pupil makes use of the computer – it is not operated solely by the teacher.

Various names have been applied in the use of computers in education, though CAL (meaning either computer-aided learning or computer-assisted learning) is now the most commonly encountered term. Other acronyms include: CAI (Computer Aided Instruction), CMI (Computer Managed Instruction), CBT (Computer Based Training) etc., but their function is generally similar, and the terms are virtually interchangeable. ICAI (Intelligent Computer-Assisted Instruction) such as that described by Kong [1994] demonstrates increased flexibility and uses artificial intelligence or knowledge-based techniques to determine the form that the instruction will take, based on an analysis of the student’s performance. (This research is of interest because although it still involves a predominantly one-way traffic of information, from computer to user, programming languages which enable a computer to make complex decisions and calculations while displaying multimedia resources may well be useful in the development of effective IEM software.)

Not all CAL software has a multimedia element; some educational software of considerable utility predates multimedia computing. However, as Chapter 3 described, human memory performs remarkably well when presented with images and vivid experiences. Thus, education is seen as an area of computing where a multimedia capability may offer particular benefits. This chapter reviews some of the significant

features of CAL and IEM software, looking for correlation between successful implementations and the principles suggested by learning theory.

4.1 A Background to CAL

As early as 1925, the first 'autoinstruction' machine was created. This mechanical device showed a series of multiple choice question cards, requiring that a student select one of four answers by pressing corresponding buttons. Operating in a variety of modes, wrong answers might require the student to repeat the exercise, or performance could be recorded for later reference by the tutor [Pressey, 1960].

In the 1950s, attempts were made to exploit this technology, which was named programmed learning. As electronic mechanisms replaced mechanical and electromechanical devices throughout that decade, it was a natural step to experiment with programmed learning presented by a computer. In 1958, a program to teach binary arithmetic was written by staff at IBM [Hudson, 1984]. By 1965, simple CAL applications were well established [Rushby, 1983].

When computers first appeared in educational institutions they existed for research purposes, later for use in teaching relating to the operation of the machines themselves, and only more recently for general computer-based learning. The 'Nelson Report' [Nelson, 1983] illustrates this change in emphasis, when compared to the report of the Joint Working Party on Teaching Computing in Universities [University Grants Committee, 1970]. The earlier report concerns a need for training in computer programming and operation skills, whereas more recent studies focus on using computers to assist teaching and learning in fields not related to computing.

Naturally, the early software was not visually impressive. However, by the mid 1980's the technologies available included interactive videodisk. The Domesday Project encouraged schools to contribute information about their area to a collaborative project meant to create a gazetteer of Great Britain, on the nine hundredth anniversary of the creation of the original document, created in the wake of the Norman Conquest. The project was completed successfully, but hardware for accessing Domesday Project data was expensive, limiting the number of platforms that could be purchased. The equipment price, the limited number of titles available in videodisk format, and difficulties in originating new material made the adoption of videodisk something of a 'false start' towards IEM in schools, with the systems remaining as occasionally used novelties [Travis and Jones, 1997].

4.2 Anticipated Benefits of CAL

Great things are expected of CAL, and some reports have been encouraging, showing that most students enjoy computerised instruction, and can in some cases learn more than from traditional methods. Beevers et al [1991] reported that their CALM (Computer Aided Learning in Mathematics) software, used to teach a calculus module to first year Mechanical Engineering students, resulted in those students scoring the best group performance. Students who were polled reported that they looked forward to the computer tutorials, and this could be seen in the form of better attendance. Shim [1992] reports that in tests at the University of Illinois, chemistry students performing laboratory experiments on videodisc scored better when tested than students who did the same experiments conventionally.

Reasons for these successes can be surmised, though it may be difficult to prove exactly what is happening in such cases. One of the potential advantages attributed to CAL is that it allows students the freedom to operate at their own pace; while the computer is not as adaptable as a human tutor, its use does at least create a favourable pupil/teacher ratio. Each student or group at a computer has their own instructor, so if the software is sufficiently flexible they may be able to concentrate on problem areas, proceeding only when ready, and keeping the learning activity within the 'flow channel' (described in Section 3.4.2). Atkinson et al, [1993] refer to an experiment where a group of children received daily instruction from 'CAI' (Computer Aided Instruction) software to improve their reading skills. Different materials and instructions were given to each student, based on the difficulty they were having at a given point in the curriculum. At the end of the study, children who had made use of CAI scored significantly higher than a control group who were taught conventionally. Nicholson & Simpson [1992] found that in addition to seeming more effective, their prototype IEM courses required less delivery time than traditional teaching, while Beevers et al [1991] found that the most common comment received in student feedback concerned the "impersonal nature of the computer tests which allowed students to make mistakes and not feel embarrassed". Clearly, CAL software has been seen to be valuable in some circumstances, once created.

4.3 Quizzes in CAL

One of the important stages in the learning process, as identified in the previous chapter, is reinforcement. Quizzes, presented as part of a CAL session, require the student to demonstrate what they have learned. Several authors including Pressey [1960] and

Wheldall [1983] have commented that delay between behaviour and reward significantly reduces the effectiveness of the reinforcer, but responses to questions embedded in CAL software can be 'marked' instantly.

In rapidly identifying the answers each student gives as right or wrong, CAL may be superior to conventional instruction, though Cohen [1985] warns that simple responses such as "correct" or "wrong" are only half the story. Students want to know their results, but knowledge of results (KR) does nothing to correct any wrong answers which were obtained. Informational feedback (IF) allows the student to remedy their error by providing additional information. A quiz in a piece of CAL software can provide KR with ease. IF, based specifically on the performance of the individual (or group sharing a computer), can be given but the software must be much more complex.

In successfully answering questions in an on-line quiz, the student might be 'rewarded' in one of three ways; by being told they have attained a good score (KR), by being allowed to progress to some new section with new information, or by being released because the learning exercise is now complete. It will be noted that these rewards are largely intangible. Naturally, it is difficult for a CAL system to supply anything more than simple praise and perhaps progression to a new section – though Pressey [1960] and Holland [1962] both experimented with 'autoinstruction' devices built to deliver a piece of candy automatically if the student achieved a certain level of performance!

With IEM, quizzes can include questions which are represented in the most appropriate medium. For example, a training application might ask staff "Is this acceptable?", showing a photograph of a part at the quality control stage. It is reasonable to expect that the value of a lesson is increased as greater realism is attained, through multimedia.

For the purposes of research into the provision of effective CAL resources, attainment tests embedded within CAL software offer the possibility of gathering data informally in a situation where the testing activity is also seen as being beneficial to the learner by confirming that they have understood.

4.3.1 Shortcomings of Quizzes in CAL

When a CAL program employs a built-in quiz to measure the amount a student has learned and determine if they can move on to a new module, it takes advantage of the tireless patience of the computerised instructor. The student can repeat the learning process as many times as necessary. In principle, not allowing a student to progress until fundamentals are mastered is a good idea, but a limitation of CAL software is demonstrated; where a good human tutor giving remedial instruction would try different teaching methods and give different examples until something to which the struggling

student could relate was found, few CAL packages are constructed to offer alternative teaching approaches. Capabilities such as this would increase the complexity of the software package, and hence its development effort, cost and final size. As a result, many CAL packages simply display the same content when an informative section is revisited (Appendix 1 describes some typical modes of navigation through multimedia software).

Many CAL quizzes feature a fixed set of questions. It is possible to have a quiz in which the learner is given a randomly selected set of questions from a larger store, but this naturally complicates the software, and it is hard to be 'fair' to a class of students if they are tested on different questions. In any event, most quizzes will test only a sample of the total information provided in the module. Students who are required to attain a certain mark in the quiz before progressing know which questions they got wrong, and as they repeat the exercise will tend to watch out for only the information they need. Since they are now memorising information specifically for the purpose of getting through a quiz which is just a few minutes away, only shallow learning is taking place. Bradshaw [1997] identifies this as rote learning, likely to be of little value in the future, as opposed to meaningful learning which would conform to the requirements of Ausubel [1968] given in the previous chapter (Section 3.2.4).

Repeated quizzes can have other problems; if some of the questions in the quiz are in the form of multiple choices an increasing proportion of these can be guessed each time the quiz is repeated. Thus, a student can ultimately score a pass mark despite the fact that they have learned little during the enforced repetitions.

To some extent, CAL has yet to come of age in that marks in a computer-based exercise are seldom used for formal assessment purposes. Difficulties centre around the ethics of assessing a student in a medium which may be subject to network delays, crashes and the other problems which can occur when using computers. Another doubt concerns the ability to guarantee the identity of a student who submits work via a computer system, there being many ways for students to share answers, or log into a computer terminal and then leave someone else to do the work. In consequence, CAL is often left as a remedial 'safety net' for optional use by students rather than an integral part of the syllabus.

4.4 CAL or IEM?

CAL is essentially interactive education in a computer-based medium, so IEM is different only in that instruction is given using multiple media. As the projects identified in Section 4.2 prove, CAL packages were producing encouraging results long before

multimedia was widely available, so it is necessary to ask if multimedia is valuable in this area. Multimedia software is expensive to create, and demands an expensive computer platform for playback, so the technology might actually be seen as an obstacle to effective computer-based learning. Learning theories discussed in the previous chapter imply that multimedia is superior because it allows attention-grabbing techniques to be employed, but this does not guarantee that a CAL programme will be improved sufficiently by multimedia to justify the increased cost of its provision.

A good CAL resource will not necessarily be that which makes the greatest demands on the computer hardware. Ultimately, a piece of CAL software may still be a valuable information resource even if it is not tremendously attractive, providing it is appropriately targeted. Some quite old, pre-multimedia pieces of software continue to be useful in the classroom. These CAL packages were understandably limited by the hardware of the time, but were effective.

Logo is an excellent example of a piece of teaching software that benefits specifically from its simplicity. A Logo exercise involves entering a sequence of instructions which the computer then uses to move a 'turtle' which draws a line behind it as it moves to produce a diagram. This is normally seen on the screen, as the example in Figure 4.1 shows, though it is also possible to control a simple robot (a 'floor turtle') via an interface. Instructions begin with simple commands such as 'forward 100' or 'right 90', with which the novice programmer can experiment. It is then possible to move on to more complex commands.

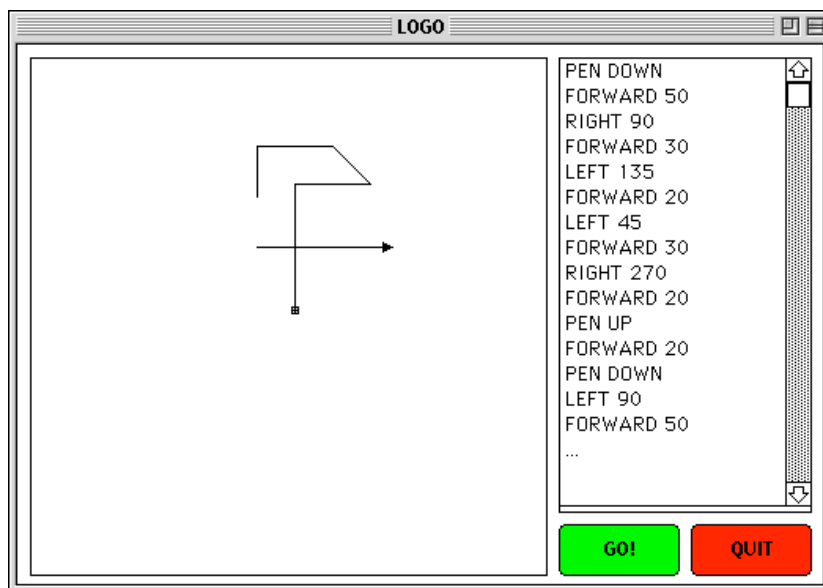


Figure 4.1: Screen display during a Logo session

New commands involving loops, variables and logical statements are all steadily introduced, and even object-oriented programming is possible [Abelson and Abelson, 1992]. The quality of the software is apparent in that young users can get to grips with the simple interface, which still manages to offer an activity that is enjoyable to students of greater ability.

Logo is visually unimpressive by modern standards, but offers the user a much greater degree of freedom than many of the more recent multimedia packages that employ a 'retrieve and display' approach. Much of the content of modern multimedia packages is in the form of large files such as images or video clips which the user can view, but cannot change. With Logo, no images are stored for display to the user, but the programmer is free to use the software to create potentially any diagram. Thus, two key features of this successful CAL package can be identified:

Freedom	The user is placed in a learning situation and empowered to make their own choices on how to proceed (a Logo class exercise could require the students to draw any picture they like).
Ownership	What the user sees on screen is a result of their own input. At all times, the user stays involved as closely as possible with the learning process.

Similar to Logo, if rather more complex, is 'Karel the Robot' which teaches elementary programming in Pascal [Pattis, 1981]. The robot begins with a very limited command set, but by working through a series of exercises, new commands are defined and employed which eventually have the robot picking his way through mazes and solving complex problems. In learning with either Karel or Logo the student finds simplicity at first, but moves on to learn an approach which will be employed in any subsequent programming which is undertaken.

4.4.1 Anticipated Benefits of Multimedia in CAL

It has been seen that the use of CAL can be advantageous, but learning theory suggests that multimedia has the potential to further increase the effectiveness of learning by supplying particularly vivid lessons, using media when they are most appropriate to get attention and promote effective coding. Where human instruction might be supplemented by showing slides or videos, giving out printed material, and suggesting where further information might be found, multimedia CAL presents an integrated set of information resources such that everything can be seen on a single machine under the

control of the microprocessor. Video clips, graphics and animations can be employed as rewards, motivating the learner towards further progress.

In the workshop or laboratory, conventional demonstrations are sometimes unsuccessful as group numbers can often be too large for everyone to see and participate fully. Numbers and safety considerations sometimes make it impossible to offer hands-on experience to every student. The use of a video film is one potential remedy, either a commercial title or something prepared in-house. Even if there is no great degree of filming expertise available, a few minutes of plain video can offer advantages, such as showing the view from a microscope to a whole class simultaneously. This allows key points to be made, and questions to be raised. Possibly, a strength of multimedia technology comes from the integrated nature of the various media; there is a capability to present very short clips of video to make specific points; in a normal classroom situation it would be impractical to display a video that lasted just ten seconds, yet such media are often found within effective IEM software.

4.5 Adaptive CAL

It has been seen that CAL software can be designed such that a lesson can proceed at any pace, simply by allowing the user to advance from screen to screen when they feel ready. Equally, students who struggle can spend time on remedial work until they reach a desired level, and the students who progress rapidly will be allowed access to new material. However, there are other differences between learners, in addition to their level of aptitude for a subject.

If a teaching function is to be performed by computer software, it must somehow allow for a wide variety of preferred learning styles. It is possible to categorise learning style, for example with the Myers-Briggs Type Indicator (MBTI). This measures personality on four different scales, as shown in Table 4.1. A total of sixteen different combinations is offered, cataloguing all individuals as having one personality type from those available; ESTJ, ISTJ, ISFJ, ESFJ, ISTP, ESTP, ESFP, ISFP, ENTJ, INTJ, INFJ, ENFJ, INTP, ENTP, ENFP and INFP. Profiles are normally found by self-testing, a process which takes about 40 minutes and involves word association [Myers and McCauley, 1985].

Characteristic	Abbr.	Characteristic	Abbr.
introvert	I	...or... extrovert	E
intuitive	N	...or... sensing	S
thinking	T	...or... feeling	F
perceiving	P	...or... judging	J

Table 4.1: The four psychometric scales used in the Myers–Briggs Type Indicator

Cross et al [1996] used the MBTI to identify the existence of preferred learning styles among people within the same profession. Since a person’s profile indicates the form of information with which he or she is most comfortable, it becomes possible to specify the most suitable approach for a CAL package, based on its subject area.

Despite finding correlations between occupational group and preferred learning style, Cross et al [1996] did not support the idea of using a profile to determine a single teaching style for everyone in an occupational group. It should also be noted that a correlation between occupational group (a person’s profession or their course of study) and a preferred learning style does not imply a correlation between a subject and a preferred learning style because students on different courses might study the same module, but learn most effectively in different ways.

As a result, it may be considered necessary to build CAL software which supports a variety of learning styles, or even one which actively adapts its interface to suit the individual user’s preferences. While it is not proposed to build full personality profiling such as the MBTI test into CAL software, it would be possible to make information accessible in several different ways and leave the user to select the form with which they are most comfortable.

Cross et al [1996] simplified the MBTI profile by concentrating on just the scales of intuitive/sensing and thinking/feeling, since these two were believed to have the greatest effect on preferred learning style. This left the four possible combinations of intuitive thinking, intuitive feeling, sensing thinking and sensing feeling. Strategies could then be defined for the best method of information delivery for learners in each of these profile groups.

ESTJ 9.9	ISTP 6	ENTJ 3.9	INTP 2.1
ISTJ 19.1	ESTP 4	INTJ 3.3	ENTP 2.1
ISFJ 15.6	ESFP 5.4	INFJ 3	ENFP 3.7
ESFJ 9.1	ISFP 6.7	ENFJ 2.3	INFP 3.8

Figure 4.2: Myers-Briggs type distribution for the general population [Cross et al, 1996]

ESTJ 13.2	ISTP 5.7	ENTJ 10.1	INTP 8
ISTJ 14.9	ESTP 3.6	INTJ 10.4	ENTP 7.2
ISFJ 4	ESFP 2	INFJ 2.5	ENFP 3.7
ESFJ 3.8	ISFP 2.9	ENFJ 2.3	INFP 4.5

Figure 4.3: Myers-Briggs type distribution for engineering students [Cross et al, 1996]

ESTJ 4.2	ISTP 2.8	ENTJ 9.9	INTP 5.6
ISTJ 1.4	ESTP 2.8	INTJ 5.6	ENTP 26.8
ISFJ 1.4	ESFP 7	INFJ 2.8	ENFP 15.5
ESFJ 0	ISFP 1.4	ENFJ 5.6	INFP 7

Figure 4.4: Myers-Briggs type distribution for art-based design students [Cross et al, 1996]

Trainor et al [1994] characterised a sample of engineering undergraduates, using the groupings sensing judging, sensing perceiving, intuitive feeling and intuitive thinking. A pair of CAL programmes were created which taught a lesson in robot programming in two different ways, one starting with theory and moving on to a practical task, and the other starting with the practical task and making use of theory as required via help buttons. Students were mixed randomly, not allowed to select their preferred version of the lesson, and testing was carried out to see if there was a correlation between the reaction of a student with a known MBTI profile type and the software they were required to use. The experiment, with a sample size of 115 students, did not yield conclusive results, though “subjective comments offered by students were encouraging”.

In addition to simply offering access in a variety of learning modes, published results exist for MBTI profiles on various groups within the population, and allow a profile of the typical learner for a particular subject to be considered. Figures 4.2 – 4.4 show MBTI profiles for the population as a whole, for engineers and for art-based design students. Percentile values (and darker backgrounds) indicate the proportion of the test group in each category. The designer of CAL software is left with a dilemma; to target a piece of software to suit the typical learner in the field, or support several different learning styles which may help a student who is struggling with the conventional methods of instruction.

In the distribution of personality types undertaking the two different courses of study, patterns can be seen. It is implied that the majority of student engineers identify with thinking rather than feeling. In the MBTI this means they are more apt for analytical, logical work requiring detached impersonality. Teaching examples should be based on physical things and products, standards and theories. Students of art-based design are categorised primarily by intuition, concerning themselves with what might be possible and the generation of alternative viewpoints. Teaching examples should refer to relationships and associations.

In the specification of informative multimedia tools and environments for construction industry professionals, Powell and Newland [1994] categorised designers using four categories; dynamic, focused, rigorous and contemplative learners (Figure 4.5). Though MBTI was not the method employed, the groups identified are analogous with the subsections of a group which might be drawn with that method. In MBTI terms, dynamic learners are NF, focused learners are SF, rigorous learners are NT and contemplative learners are ST. The strategy employed by the subject in a learning situation was found to be highly inflexible, with their world-view affecting the ways in which they

expected to receive useful information, each with consequences for the author of IEM software.

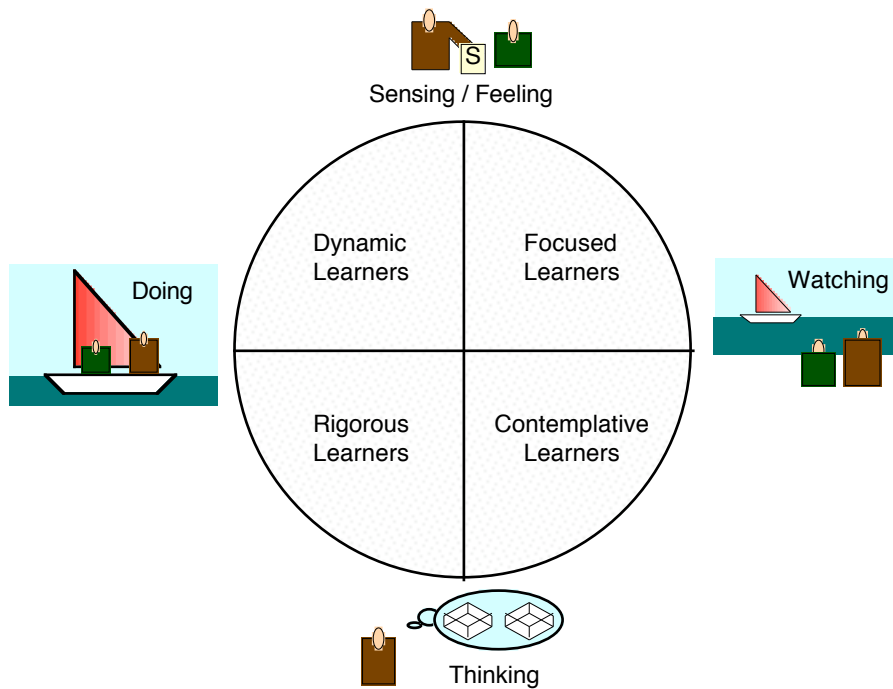


Figure 4.5: Four modes of interaction with the environment [Powell and Newland, 1994]

CAL software may well be more effective if it allows for some variation in learning style. Unfortunately, the required development effort will increase with any attempt to make the software more flexible. Trainor et al [1994] pointed out that developing software with special responses for sixteen different personality types would be unwieldy (and in any case, software design for a large number of subtly distinct personality types would be highly subjective and influenced by the programmer's own personality type). It may be more practical to limit special accommodation of preferred learning style to a smaller number of more broadly defined personality types such as those in Powell and Newland's [1994] model.

4.6 Interactivity and IEM

Interaction is a key feature of CAL software such as Logo (see Section 4.4), allowing users to experiment, test what they have learned and choose what they want to do. In addition to their personality type, people differ in their learning needs (and interests).

IEM software can accommodate these differing needs if it allows the learner enough opportunities to supply input and make choices which control the learning process.

IEM software can be characterised by its location on a scale of interactivity. The computer can be programmed to present one of a number of different operating regimes, and the choice made will have an impact on development effort, applicability of the software to various audiences, and how memorable the experience will be.

Interactivity need not be confined simply to selecting information to view; in fact this constitutes a fairly low level of flexibility. On-line simulations and adventures – what Quinn [1994] calls ‘exploratory micro-worlds’ allow the learner to experiment within the software environment in an effort to understand a system or process. Quinn [1994] and Katz and Offir [1994] both found educational games of this kind to be intrinsically motivating, allowing effective learning.

Schwier and Misanchuk [1993] offer a ‘taxonomy of multimedia interaction’ with four levels. Level one is the most meagre, requiring only the simplest videodisk hardware. The video which is presented is entirely linear, but can start and stop at predesignated frames, allowing ‘chapters’ which are of interest to be selected.

Level two involves the additional use of control codes, prerecorded into a redundant audio channel. A microprocessor built into the videodisk player reads these programs, presenting the viewer with choices or quizzes. Naturally this interactivity increases the development effort required for the videodisk programme. Also, multiple formats must be catered for, since the two main videodisk players (Philips and Sony) use different control code protocols.

Level three in Schwier and Misanchuk’s model might be considered ‘true’ multimedia, making use of a microcomputer. Early systems were ‘two screen’, presenting the user with choices or quizzes on the computer’s monitor, and then exercising control over a videodisk player to display the relevant film on a separate screen. Later, it became common to display all the media on a single screen. At level three, the versatility of the computer is combined with the quality of videodisk images. Of course, the power of the computer may be used to a greater or lesser degree, depending on the programming. Schwier and Misanchuk [1993] categorise the union of the two technologies in three ways. In ‘Computer as pathfinder’, video is used predominantly but the computer presents choices to the user. In ‘Computer as partner’, instruction takes place in the form of both computer generated interaction and video sequences, and in ‘Computer as pedagogue’ the computer’s demands on the user are dominant and the videodisk medium is used to a lesser degree.

In Schwier and Misanchuk's [1993] model, level four referred to the future. Emerging technologies such as virtual reality graphics, touch screen interfacing and speech recognition were the anticipated features. In truth, these technologies represent superior user interfacing, not interactivity. Communication between user and computer may be easier, but the underlying software itself must still be programmed to invite and support user input. Virtual reality may form the basis of an interactive environment, if appropriately programmed, but technologies such as touch-screen interfacing are generally confined to simple kiosk applications with a low degree of interactivity.

Moving on from the microcomputer controlled set-up described as level three by looking at what is possible in software development, rather than emerging technology, it can be seen that a wide range of levels of interaction between the user and computer can be found. Current systems typically make use of a built-in CD-Rom drive rather than a videodisk player (this newer technology being much cheaper though with somewhat less attractive video picture quality), but in either case the capability exists to run interactive software. This may take the form of user-controlled browsing through multimedia information, quizzes, simulations or other interactive elements. Such software should produce more effective learning experiences, according to Buzan's [1993] requirements for successful recall (see Section 3.4.3), e.g. "Items which are of particular interest to the learner".

4.7 Development Effort for IEM Software

In Chapter 1 it was suggested that the development effort inherent in the creation of IEM software has been a major obstacle to wider exploitation of the technology. As Marshall et al [1994] showed, development requirements for a piece of software can be forecast at the specification stage. Significant factors identified included the volume of non-text media to be used and some constraints under which the developer might operate.

The development effort required for multimedia software is high, compared to that for a conventional application. The graphic user interface and any interactive elements require specific programming, and additional digital media such as sound tracks and video clips must be created or obtained. This is a particularly significant problem for educational multimedia, where the target 'market' for the software may not be large enough to justify the outlay required.

The development requirement will rise with increases in interactivity, programme scope, media quality and other factors. Development effort can be expressed as a ratio of

time spent in development against the anticipated playback time for one student. When quantified, this ratio can be quite disturbing. Optimistic estimates place it in the region of 50:1 [Avner, 1988], but an exact figure would depend on the aims of the software produced, and ratios of up to 800:1 have been reported [Golas, 1993]. A common consensus for development time to playback time is 300:1 [Marshall et al, 1994].

To put this figure into context, the 'computerisation' of a single tutorial will be considered. For an hour-long tutorial, multimedia courseware might be required which will be viewed for around 40 minutes, the rest of the time being taken up in getting the users started, and concluding the session satisfactorily. At three hundred hours' development effort for a single hour of playback, the academic wanting to create the computerised tutorial described would need to set aside 200 hours for development work. Effectively, he or she would have to work exclusively on its development for about six weeks. Of course, it is unlikely that such an uninterrupted block of time could be found. Assuming it is no less efficient to carry out the software development gradually, the same piece of courseware could be created in four hours' work per week for a year. Even if this time can be found, this will produce enough multimedia courseware for only one tutorial, when ten or more such tutorials might be required for use in a single subject module.

The scope of the software created (the issues it addresses and skills it imparts) is thus limited by the development effort required, possibly cut back to just the areas required for the syllabus of the target user. However, a capability to select topics which are of interest (or which are going to be most useful) from amid a larger information store would make the software valuable to a wider audience. This form of interactivity is desirable in educational software, but demands greater development effort, in indexing the information to ensure that the user can select specific sections, and in providing enough information such that there is 'something for everyone'.

It may be possible to justify increased development effort undertaken if it increases the number of potential users of the finished product (for a commercial multimedia title), or results in a higher quality of learning experience (for a piece of in-house multimedia). The useful life of a piece of software before the information within it goes out of date may also be a factor in determining if the development process is justified.

4.7.1 Justifying Development Effort

The time and money required for multimedia production can be tolerated if the target audience is large. Again using the sample IEM tutorial where 200 hours' work (perhaps £4,000) are required to create a 40 minute programme, this might be prohibitive if the system will only be used by fifty students a year, but the time and money might be

considered to have been well invested if the system allows high quality instruction to be given to a thousand people.

In a business planning to develop a staff training programme, a calculation of training cost per person would feature fixed and variable elements. The variable cost element can be calculated as the trainee's wage and share of business overheads (or the cost in lost productivity) for the time spent training. Each trainee must also carry a share of the fixed costs which were incurred in setting up the training programme. Thus, the cost of training a person might be expressed as:

$$C = w + \frac{s}{g} + \frac{d}{n.g}$$

- where
- C = training cost for one person
 - w = cost of taking the trainee away from their work for the required time
 - d = training programme development cost
 - g = class group size for the training programme
 - n = number of groups to be put through the training programme
 - s = trainer or session supervisor's cost

Conventional classroom teaching methods will have relatively low training programme development cost element (d), but will require a skilled person to deliver the training. This could be expensive, and close supervision might be required. This would reduce the group size which is possible and thereby increase the cost per head of the training resource ($s \div g$). Large group sizes offer economies in this form of training, but may be accompanied by a reduction in the quality of learning which takes place. If a multimedia training resource is created, delivery involves less direct input from a human supervisor, and group size becomes less critical. A group size of just one person, unsupervised, is possible. The most significant cost where multimedia is employed is in the system development cost (d). Economies are found when a large total audience ($n.g$) is put through the multimedia programme. For American Airlines, multimedia allowed the training of flight attendants, reservation agents, cargo personnel and others totalling over half the organisation's employees. American Airlines considered this sufficiently important to develop their own multimedia authoring system, anticipating a 50% reduction in training time per employee [Cillo, 1991].

Though a wider audience makes justification of development cost easier, there may be difficulties in 'exporting' the software developed. Variation in machine specification can result in either a complete failure to run the software, or impaired running. Another significant problem is that the software might not be entirely appropriate for use elsewhere. Some teaching packages have had humble beginnings, in a single educational institution, created by the academic staff who are responsible for teaching the subject. Since they are specifically targeted, these packages usually achieve their initial objectives. A wasted opportunity can be seen, however, as such software is seldom found in use outside the organisation that created it. Specialisation to suit a single syllabus can be harmful in the long-term.

4.7.2 Desired Qualities in an IEM Programme

While a user might say that software should be "easy to use" etc., such requirements may be difficult to quantify. Three quantifiable measures of the quality of a piece of IEM software (for a given training task) are proposed. The desired qualities of the multimedia programme might be expressed in terms of the following:

- Efficiency with which information is retained by users
- Speed at which a given message can be conveyed
- Volume of information available within the system

The fraction of information viewed which is retained by users will clearly be important in educational applications, being a direct calculation of the amount of correct answers a student is able to give as a proportion of the total volume of information taught. This can be considered a measure of the power of the system to command the students' attention and hold it during a learning exercise. A test at the end of a session, either a computer-based quiz or conventional exam, can be used to assess the student's level of knowledge.

In safety applications, or in a diagnostic context where immediate action is required, system clarity becomes the critical factor. The purpose of such a multimedia system is not to supply information for long-term recall, but to get the user to perform a required action as quickly as possible. An early example of this kind made use of multimedia to brief and dispatch shipboard fire fighting crews [Sutcliffe and Faraday, 1994]; other applications include a kind of trouble-shooting system for computer hardware, which allowed people to diagnose and rectify simple faults without calling in an expert [Clarke et al, 1994]. However, for most educational applications, speed will not be the critical factor and the user will be able to proceed at their own pace.

The breadth and depth of information available within a multimedia information system will need to be sufficient for its users' purposes. An incomplete information resource is frustrating, but against this the inclusion of additional information might strain a development budget or exceed the technical limitations of the delivery system. Careful specification of the content at the design stage is the only way to ensure that a satisfactory system can be developed.

4.7.3 Limitations on the Development of IEM Software

In this chapter, development effort has been identified as a significant obstacle to the creation of IEM software. The required inputs which might constrain a development project include:

- Costs that will be incurred in the development process
- Time to construct the system (man hours and overall project length)
- Skills needed for development of the system as specified
- Technical limitations of the delivery system

Naturally, the cost to complete a multimedia system is a major factor, and a large element of the cost is likely to come from the amount of programming and data input required. In addition to a possible constraint in terms of the total number of hours for which personnel are available there may well be a delivery deadline to meet, and system development is not just a question of man hours, but of input from personnel with appropriate skills.

The desired qualities of a multimedia programme will also have an effect on the delivery system required, namely in terms of the storage space required (which will affect the method of distribution), and the processor speed of the computer to be used for playback.

It was stated in section 4.7.2 that the desired performance of an informative multimedia system can be specified in quantifiable terms. This might be in terms of the percentage of content which is retained after a session, the speed in seconds required to get access to urgently needed information, or the volume and nature of content included in a multimedia information system – for example, the latter might be measured in terms of word count, number of pictures, total minutes of digital video, number of exercises provided or the percentage of a syllabus supported. The constraints upon a multimedia system can also be quantified; in terms of the implementation deadline, project budget, man-hours available within various skill categories, and technical measurements such as kilobytes of RAM and disk space required for the completed IEM software.

Schneiderman [1980] illustrated this conflict of interests when he described software psychology in terms of human-centred and system-centred factors (Table 4.2). The user wants software which is easy to use, reliable and comprehensive (note that these qualities are not necessarily easy to quantify), but the developer must bear in mind the limitations imposed by the computer hardware.

Human (user)	System (developer)
Ease of use	Machine efficiency
Simplicity in learning	Storage capacity
Improved reliability	Hardware constraints
Reduced error frequency	
Enhanced user satisfaction	

Table 4.2: Human-centred and system-centred factors in software development [Schneiderman, 1980]

This view of desirable qualities versus constraints can be applied just as well to IEM as to the ‘normal’ computing applications which Schneiderman would have seen in 1980 (his ‘Simplicity in learning’ requirement refers to learning how to operate an application itself – it does not address CAL). With the exception that development cost/effort are excluded, Schneiderman’s goals and constraints offer a fair model of the influences on software design.

4.8 Conclusions

This chapter has described some of the features which might be incorporated into a piece of IEM software in an effort to facilitate effective learning, in accordance with the requirements outlined in the previous chapter. Thus, a point has been reached where it can be seen that the approach which is taken when aiming to make use of IEM will be a compromise between three influences:

- Human learning issues (preferences and limitations)
- Technical constraints on what is feasible
- Development effort and other cost related issues

In consequence, a piece of IEM software must be sufficiently comprehensive, easy to use and effective in order to appeal to a sufficient number of users such that its development cost is justified.

It is concluded that it is necessary to reach a sufficiently detailed understanding of the value of each medium and technique available to the IEM software developer such that their value is quantifiable and can be considered prior to multimedia programme development. It is proposed that this would offer an effective strategic model for the use of multimedia to achieve a given set of learning goals for a given audience. The alternative is that developments are technology-led or decisions are taken simply as a result of programming expediency. Quantifying the value and cost of each technique will be useful in justifying the time, budgetary and logistic requirements involved for multimedia software development.

In order to direct development effort and resources most effectively, it will be necessary to know the media and presentation styles which can make the most significant contribution towards learning. Experimentation towards this aim was the purpose of several pieces of experimental software, detailed in Chapters 5 and 6. In this development work, it was aimed to apply the learning theories described in the previous chapter.

While the uptake of CAL and IEM speaks for itself, the sheer volume of titles does not guarantee the quality of the software produced. Progress in IEM has at times been pushed by emerging technology rather than pulled by demands from potential users; Goble [1994], as guest editor of a journal which ran a special issue on multimedia information systems, expressed disappointment at the lack of submissions on the subject of applications rather than the technology itself. Marshall et al [1995] identified that development of CAL material can be quite haphazard, subject to cost and time over-runs. Cann [1997] suggested that few pieces of IEM software are still in use four years after their creation, further reducing the possibility of IEM software reaching a point of payback. There is a danger that IEM software produced in-house in educational institutions solves an immediate teaching need, but does not produce software which is readily exportable to other institutions.

5.0 The Application of IEM Software to Promote Design for Manufacture in the Electronics Industry

CAL and IEM software offer more possibilities than simply supporting the formal education of full-time students. If the software is sufficiently easy to use that problems are unlikely to arise, informal 'just-in-time' training becomes possible. Rather than the costly business of taking people away from their work and training them periodically, IEM allows training to take place when it is needed, pausing and resuming as required. With the computer offering an information resource right on the shop floor, the user is not placed in an unfamiliar environment which might reduce the effectiveness of the learning process, and they are able to put what they learn into effect almost immediately, which is likely to reinforce understanding.

One area in which the capabilities of IEM might be particularly useful is in the ongoing training of electronics industry design personnel. The design of electronic products is a formidable process, particularly as new products are expected to be increasingly complex, yet often also smaller, cheaper and more reliable. In making a design choice to meet one or more of these and other requirements, compromises have to be made. One aspect of product design which might be neglected is ease of manufacture; products can be designed which have the desired functionality, but which are difficult to produce in quantity. Problematic items must either be reworked, or scrapped, either strategy adding cost.

Designers often have only a limited appreciation of manufacturing issues because the creation of an electronic product is typically carried out across a network of organisations. Designers are often located away from manufacturing operations, even in a different country, so they can be unaware of the relative costs and difficulties caused by the design decisions they make.

It is necessary to give the designer an appreciation of the issues which arise as a result of design choices, but periodic formal instruction could be expensive and inefficient. Paper based documentation, even with plenty of illustrations, will not be entirely effective as it is necessary to show process equipment as it moves. Movies and animations can communicate information of this kind more easily.

This chapter gives an overview of the processes involved in the manufacture of a modern electronic product. The in-circuit testing (ICT) phase is used to illustrate how IEM software might allow people not directly involved with a process to gain an appreciation of the issues involved. Experimental software which was created to demonstrate this is described, and some implications for the development of IEM software are identified.

5.1 An Overview of PCB Manufacture

It has been stated that the design of electronic products involves compromise between a number of issues; some of these are identified here to illustrate the problem. In the electronics industry, as for any manufacturing operation, production can be identified as a sequence of operations. Put simply, the personnel at each stage are in receipt of information and/or materials, they carry out some necessary conversion function, and they pass on the results. The subsections that follow describe the design and manufacturing stages which an electronic product might undergo.

5.1.1 Design

For an electronic product, the typical PCB design process will involve interpreting a circuit diagram and producing a board layout with the desired functionality. Once the electrical functions of the product have been specified it becomes necessary to fit all the required components onto the circuit board and achieve electrical connection between them. Nowadays this task will be achieved through the use of an electronic CAD system with the operator stating where the components should be placed and then having the system do the routing work, mapping out the conducting tracks which make the required interconnections. There are many issues to consider as the designer works to place all the components, such as the shape of the casing that the board is to go into, problems with heat dissipation, electromagnetic compatibility, and keeping path lengths equal in high performance circuits.

5.1.2 Board Manufacture

Conductive tracks between components are produced by a photo-etching process. Most modern products feature multi-layer boards; sandwiches of conductive and insulating material laminated together in precise registration. This is necessary to achieve the complex interconnections required between all the component leads. Drilling is carried out to allow component leads to pass through the board, and drilled holes may be plated to connect tracking on different layers. Solder-resistant ink may be printed on areas of the board, meant to repel liquid solder, and a further printed ('ident') layer might be used to locate components during placement, manual testing or maintenance.

A good design will employ as few layers as possible, since these increase the cost of each bare board. A good design will also feature tracks of adequate width, since very narrow tracks can make board manufacture difficult, and reliability questionable.

5.1.3 Placement

Each PCB must be populated with components. Conventional components are placed by inserting their legs through holes in the board. This type is referred to as Through-hole Technology (THT). An increasing number of products use components in the smaller Surface Mount Technology (SMT) format. These components are 'onserted' either using adhesives or simply by placing them on a printed deposit of solder paste. Figure 5.1 shows the through-hole and surface mounted forms of a resistor.

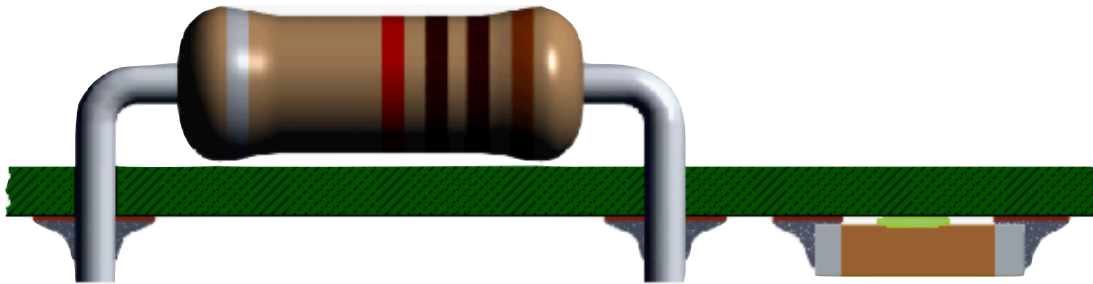


Figure 5.1: Through-hole and surface mount resistors

The vast majority of placement operations in volume production are carried out by automated machinery, though some problem components might require manual placement. A good design will allow automatic component placement as far as possible.

5.1.4 Soldering

For conventional components, electrical connection is achieved by passing the PCB over a wave of molten solder. The solder clings to component leads and copper tracking, solidifying in place as it cools. Small SMT components, glued to the underside of the PCB, can also be soldered at this time. Other SMT components on the top side of the board are attached by using heat to liquefy the alloy beads within a printed deposit of solder paste, in a separate process which is called reflow.

5.1.5 Cleaning

It is necessary to remove oxides from the PCB and components if good soldered joints are to be achieved. The flux used for this is usually corrosive. In this case, any residues must be washed from the assembly or they might shorten its life. There is some move towards fluxes which do not require cleaning, or which do not require solvents to be employed in cleaning, but these can exhibit poor performance at other manufacturing stages, particularly solder paste printing [Currie et al, 1994]. This is characteristic of the

complex inter-related processes in the electronics industry, with changes to achieve one goal often making other processes more difficult.

5.1.6 In-Circuit Testing

After the PCB assembly has been made, it is necessary to carry out In-Circuit Testing (ICT) to ensure that there are no faults in the product. This requires that probes can be attached to key points of the circuit to alter or monitor their electrical state potential. For volume production, probing is done with a custom built automated test fixture which uses vacuum or mechanical clamping to bring the PCB into contact with a number of spring loaded probes, each touching a designated test point on the surface of the PCB (Figure 5.2). An automatic in-circuit test machine can then inject waveforms into the circuit to verify that it is performing as designed. A good PCB design will allow probing to be carried out with a fixture which is easily built and reliable. A good design might also make use of integrated circuits which allow boundary scan testing, reducing the total number of test points required.

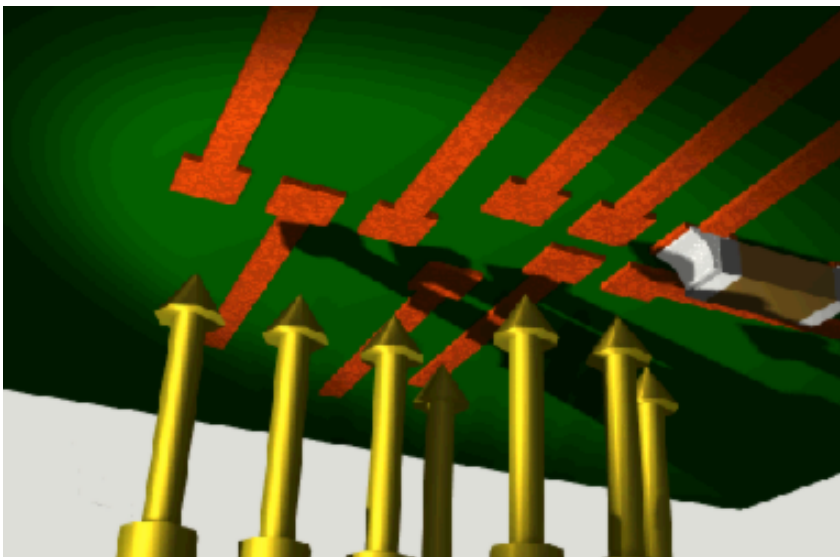


Figure 5.2: In-circuit testing concept

5.1.7 Rework

As an alternative to scrapping PCB assemblies of high value when a fault is detected, rework might be attempted. This is usually a manual task because problems tend to vary in nature from one faulty board to the next. A good design will allow rework to take place if that is the policy (as opposed to scrapping all faulty products). Ideally, faults will not occur and rework will not be necessary. Modularity reduces problems here because it is not necessary to scrap the whole product if a single board fails. However,

multiple board assemblies may be undesirable for reasons apparent at other manufacturing stages, requiring extra connectors and extra assembly operations for example.

5.1.8 Assembly

Many electronic products feature more than one PCB, each of which will undergo manufacture separately. In addition to connecting circuit boards together, assembly will usually involve casing the product. At this stage a well designed PCB assembly will be easy to handle and secure.

5.2 Case Study: Design for In-Circuit Testability

Section 5.1 showed that there are a number of separate manufacturing processes which take place once a new product has been designed. Decisions taken at the design stage might make each of these tasks easier or more difficult, and might also influence product cost, size, weight, reliability or performance.

Increasing awareness of the influence which design has on the testing stage of the PCB manufacturing process was the subject of the Design for In-Circuit Test Advisor (DICTA) project in the Institute for Design, Manufacture and Marketing at the University of Salford [Gallagher et al, 1994]. Research ultimately identified more than thirty aspects of PCB design which could impact on testability [Gallagher & Lawlor-Wright, 1995].

Gallagher [1996] describes software which took the PCB layout, produced in an electronic CAD package, and consulted a library of details on each design feature (components, via holes, test points, board edge segments and others). A three-dimensional model of the PCB assembly could thus be inferred, the physical height of each component on the probed side(s) of the board representing a potential obstruction to automated probing.

The three-dimensional model of the design was built up in ICAD, an object oriented modelling package. Each entity in the design was considered for its compliance with a set of DICTA guidelines and a report was generated in which any feature which had negative implications for testability was identified. In theory, this left the designer with a critique of the PCB layout he had defined, in the form of a printout where each section represented one of the DICTA guidelines, and any components in violation were listed. The designer could then consider altering the design for better compliance with the guidelines. Unfortunately, as with many knowledge-based systems, the reports produced were quite

terse, and used terminology which might have been unfamiliar to the PCB designers. An extract is reproduced below:

```
...
=====
Report on edge clearance for vacuum seal- Rule 3
=====
BOARD-UNDER-REVIEW
This rule reports on the preferred and absolute edge clearance

These components are in violation of rule three, Edge clearance
for vacuum seal

                Minimum Preferred Distance allowed is : 200

Comp Name Min Dist          Coor-x    Coor-y    Orientation
-----
C11        107.5d0          -5900    2000     90
U9         50.0d0             -8800    2400     0
C10        107.5d0          -5900    2650     90

                ABSOLUTE Minimum Distance allowed is : 100

These components need to be moved, to ensure a vacuum seal

Comp Name Min Dist          Coor-x    Coor-y    Orientation
-----
U9         50.0d0             -8800    2400     0
...

```

Again, the problem that designers may be unfamiliar with the automated testing equipment reduces the quality of the decisions they can make. Some rules are absolute and must not be broken, while a little flexibility is possible with others, when necessary. With clear guidelines, the designer can be made to understand the cost or risk inherent in an infringement. The extract reproduced above attempts to explain how one component must be moved, though it and two others may be placed within 0.1" of the board edge if necessary. Infringement of some DICTA rules is much more expensive than others; for even a single test point to be located on the 'top' side of the PCB would massively increase the complexity of the fixture, and reduce its reliability. Thus, compliance with some guidelines is more important than others and identifying a component or feature in violation of a rule is not enough if the designer does not understand why the rule exists.

As stated in section 5.1.8, automated in-circuit testing requires reliable physical contact between a PCB and its test fixture. The layout of electronic components on the PCB is significant because each component represents a potential obstruction to the probes. PCBs must also be suitable for handling and being held accurately in place. A product which otherwise meets customer and manufacturing requirements may need an intricate test fixture which is more expensive and less reliable – and an imperfect fixture could report faults where none exist, increasing scrap and/or rework costs. Turino [1990] detailed design choices and features which would facilitate testability, and [1991] compared the costs to find and rectify a fault in a board with basic design for test (DFT) features against those for a basic board, finding that DFT saved time and money. Bullock [1987] stated that a PCB must be designed to accommodate testing if that activity were to be carried out successfully.

5.2.1 Existing Design for Testability Guidelines

Some companies have recognised that there is a product testability problem and have circulated guidelines for their designers. One such set of guidelines was seen on a visit to an ICL facility [Hargreaves, unpublished]. The Printed Circuit Interconnect Federation [1996] produced a booklet 'The Enlightenment of Albert' with DTI funding to encourage personnel to consider manufacturing issues, including testing. Despite these efforts, ICT fixture manufacturers continue to receive PCB designs with many fundamental accessibility problems, such as very small test points, test points located too close to components, components too close to each other or too near the edge of the board. While it is possible to meet the test requirements for such products, the fixture supplied will be more expensive, less reliable, and will take longer to build [Gallagher et al, 1994].

To supply designers with clear testability advice, a multimedia package subsequently known as the DICTA Information System (DIS) was developed which could be integrated into the design environment. Section 5.3 details the options considered and decisions taken in the implementation of that system.

5.3 The DICTA Information System

The Dicta Information System (DIS) was developed to appeal directly to PCB designers, making multimedia information available not under a synthetic, periodic training regime but placed directly within the design environment itself. Sections 5.4 – 5.7 describe programming work undertaken to present testability guidelines, first as on-line

documentation and then making increasing use of interactivity. The authoring software employed in each case is described in Appendix 2.

With most of the testability guidelines already gathered as an early part of the DICTA project (see Appendix 3), the first stage of development in this research was to collect or generate visual media which would support the existing text-based rule document. It was found that very little suitable published material existed, and copyright was reserved on all the published photographs found. Later a trip to the test fixture manufacturer, Everett Charles Technologies, allowed an opportunity to photograph some test equipment. Meanwhile, diagrams were drawn to support each of the guidelines.

The decision to use an IEM approach to promote design for testability was later borne out to some extent by the creation of a new version of the DTI funded booklet 'The Enlightenment of Albert' [Printed Circuit Interconnect Federation, 1996], this time supplied on CD-Rom [Printed Circuit Interconnect Federation, 1997] and if not tremendously interactive, at least employing multiple media.

5.4 The DICTA Slideshow under Microsoft PowerPoint

Initial work towards the creation of a multimedia information system was carried out using Microsoft PowerPoint, resulting in the production of a rolling slideshow representing the design advice advocated in the DICTA guidelines. Each rule was accompanied by a graphic, for emphasis and to illustrate the manufacturing equipment to designers who may have been unfamiliar with it. Simple animations were achieved by showing several slightly different screens in rapid succession which together give an illusion of motion (Figure 5.3).

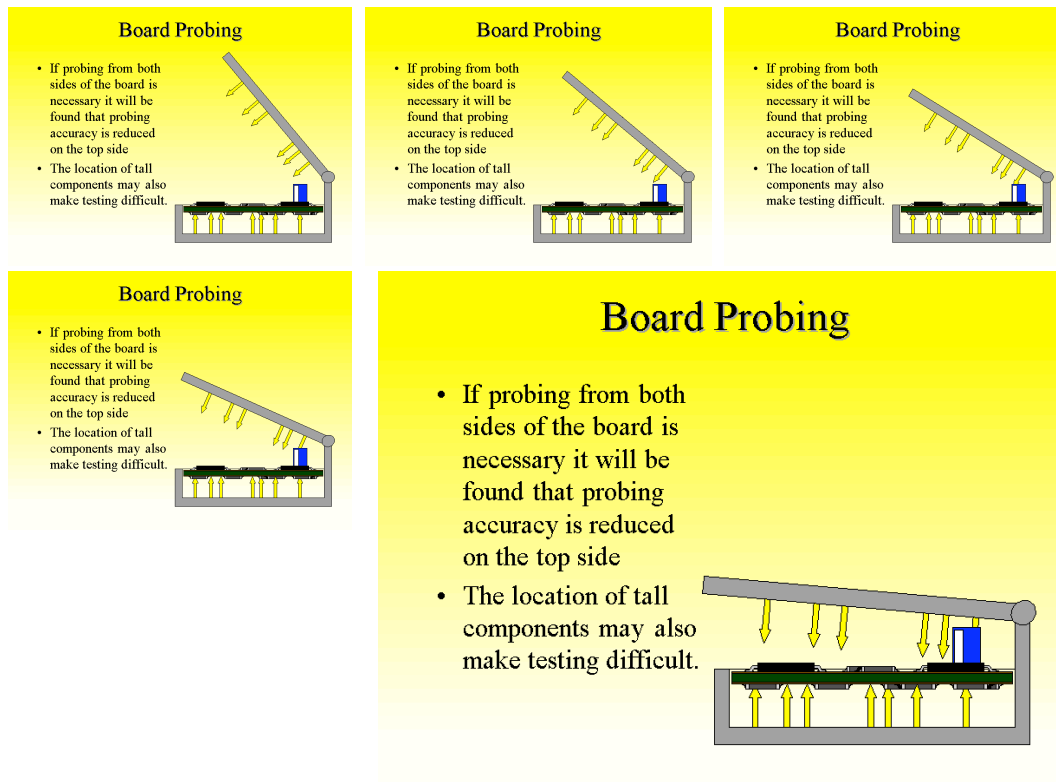


Figure 5.3: Multiple images used to form an animation in the DICTA Slideshow

The PowerPoint slideshow's simple animations and transition effects between pages began to illustrate an advantage offered by multimedia technology in that this approach was eye-catching. It was important for the information to be clear and attractive if it was to be incorporated into the existing PCB design environment. Even so, the slideshow does not qualify as multimedia by the definition given in Chapter 1 since there was no real interactivity. The document was entirely linear, advancing from page to page after a short pause or when the mouse button was clicked. Though failing to qualify as multimedia, the prototype showed some fundamental advantages that would be found with computerised design for testability information;

- Information employing colour diagrams could be distributed for a much lower investment than would be required to create an equivalent paper document.
- The guidelines could be disseminated easily; on a single floppy disk, through e-mail, or by downloading over a network.
- Crude animations showed the operation of test equipment where descriptive text would have been cumbersome.

5.5 The DICTA Information System under HiPWorks

Having had some favourable responses to the PowerPoint Slideshow when it was demonstrated to electronics industry managers, further integration of testability advice into the design environment was desired. Since much PCB design work is done on UNIX-based CAD systems, it was preferable to present a version of the DIS on that platform. HiPWorks, a multimedia authoring package from Integrated Solutions Ltd. was used.

Essentially, HiPWorks is a program allowing the construction of a graphic user interface as a plug-in to an existing language which normally operates in a text-based mode only. That language is Poplog, a combination of Pop-11, Prolog and Lisp. Poplog is potentially very powerful, allowing the creation of expert systems and object oriented programming. Unfortunately, while Poplog was found to be ideal for some tasks, the control of an IEM programme is not one of them. Programming to handle user interaction and multimedia resources was cumbersome, and some tasks which are simple for other packages were very difficult or impossible, an inability to support hypertext links or play video clips being particularly significant shortcomings.

Problems with bugs were encountered which occasionally resulted in the system 'hanging', and loss of data sometimes occurred when a save operation would write an empty file. Attempts to build multimedia capabilities into the DICTA system using the HiPWorks authoring software were disappointing. As a concept, the HiPWorks DIS was superior to the PowerPoint slideshow in that it was interactive rather than simply linear, but progress was very slow.

IEM software promoting design for testability which could be viewed on designers' own workstations would have been ideal, but none of the multimedia authoring languages for UNIX systems seemed appropriate. Appendix 2 details commercial authoring packages available at the time (ca. 1995), explaining the software choices made. The only well established alternative to HiPWorks which ran under UNIX was IconAuthor, which was investigated, but found to lack many of the features of the PC version when running under UNIX.

Work centred instead on a prototype created in HyperCard, an authoring language which runs on Apple Macintosh computers. This is not a common computer in UK manufacturing industry but the stable, well-established authoring language allowed prototyping to proceed at speed. In a last attempt to present testability advice in multiple media on UNIX machines, some work was done (in conjunction with a colleague) to make the DICTA analysis system produce its testability reports in an HTML format so these could be displayed by a 'Web browser on a variety of platforms. For graphics, this system used the existing screen displays from the PowerPoint slideshow. It was seen that

comments on a PCB's testability could be expressed and supported with graphics on a UNIX machine, but the level of interactivity that could be achieved with HTML code was insufficient for the purposes of the DICTA project.

5.6 The DICTA Information System under HyperCard

The PowerPoint slideshow on testability had demonstrated that images and animations were potentially useful in conveying a design for manufacture message, but it did little else to promote the notion that testability was a necessary aspect of a quality design. It was an on-line document which the designer could choose to consult, but which did nothing to actively encourage the designer to think of testability during the design stage.

Fischer et al, [1993] suggest that critiquing systems are most effective when placed in the design environment itself, and Cann [1997] maintains that multimedia is most effective in educational activities that already require the use of a computer, such as simulation. Where a computer is already being employed in the PCB design process, the user should be sufficiently computer-literate to make use of an IEM module. Since, as section 5.2 explained, it is impossible to apply absolute rules on testability and the designer must be relied upon to make an informed decision, IEM is an ideal way to make the necessary information available. It was, therefore, a goal to integrate multimedia advice with the existing DICTA analysis software.

A program was written which imported the text files produced by the DICTA critiquing system. Pattern matching (looking for key words or phrases within the critique) was used to identify when guidelines had been breached. Pages of information existed in the IEM software which referred to each rule and the consequences of its violation. These were automatically 'bookmarked' if the corresponding imported file indicated that a rule had been infringed, in effect allowing the DIS to display multimedia resources which were selected as being of current relevance. Figure 5.4 shows a screen shot from the HyperCard prototype. HyperCard allowed considerably more interactivity than had been possible in the PowerPoint slideshow. The programme developed was essentially an 'electronic book about electronics', where a designer could read pages of information sequentially or jump to areas of particular interest. Many of the pages in the HyperCard DIS were analogous to those which had appeared in the earlier PowerPoint Slideshow though there were other, supporting pages of a glossarial nature.

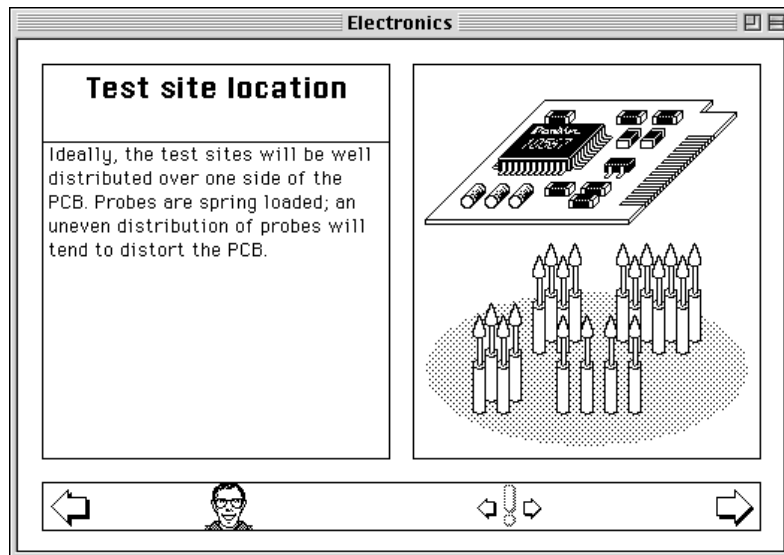


Figure 5.4: Screen display from the HyperCard DIS prototype

By launching the multimedia system and allowing it to scan the critiquing file quickly, the user was not required to read it in detail to get an overview of any testability problems that are likely to arise with the design. The multimedia information effectively forms a supplement to the feedback section of the critiquing process. Figure 5.5 shows the architecture of the DICTA environment; the designer's inputs to a CAD system are translated into a product model which can be analysed for testability by the existing DICTA software. The resulting critique can either be interpreted 'raw', or run through the DIS and viewed in an IEM format.

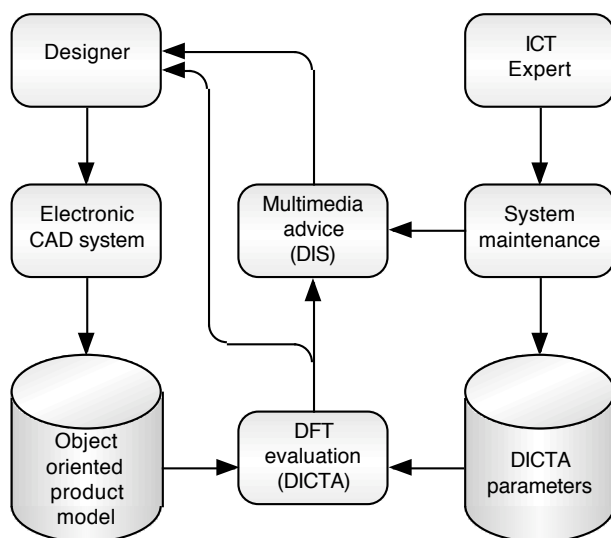


Figure 5.5: DICTA environment architecture

Each page of the HyperCard DIS contains text and graphics. Pages relating directly to the DICTA design rules also have a pop-up field which is automatically populated with a list of any components in violation of that rule, together with their location and any other relevant details. Thus, the designer is informed of anticipated testability problems while he is in the best place to learn more. The designer can also view an index page containing a set of buttons identifying each rule which has been violated so that those pages in particular can be visited. In this way, the DIS saves the designer time, reading through the whole of the text-based report and offering a summary in a few seconds.

It was an aim throughout to keep the system simple and attractive. In effect, consulting the system was an additional task to be carried out by the designer, so emphasis was placed on ensuring that the software was reliable and pleasant to use.

5.6.1 Dynamic Generation of Graphical Content for the DIS

In addition to providing pages of design rules with generic illustrations, an experimental module within the DIS generated screen displays which reproduced the current PCB design itself (Figure 5.6). Reasoning that simple lists of components in violation were less likely to provoke successful redesign than a graphical, interactive approach, it was hoped that the graphical nature of the report would demand attention more effectively, and reduce ambiguity. Even so, this is a compromise; ideally the testability critiquing would be carried out within the electronic CAD system itself, presenting clear design advice. In the absence of such a CAD system, this module of the DIS attempted to bring the board layout into the multimedia environment.

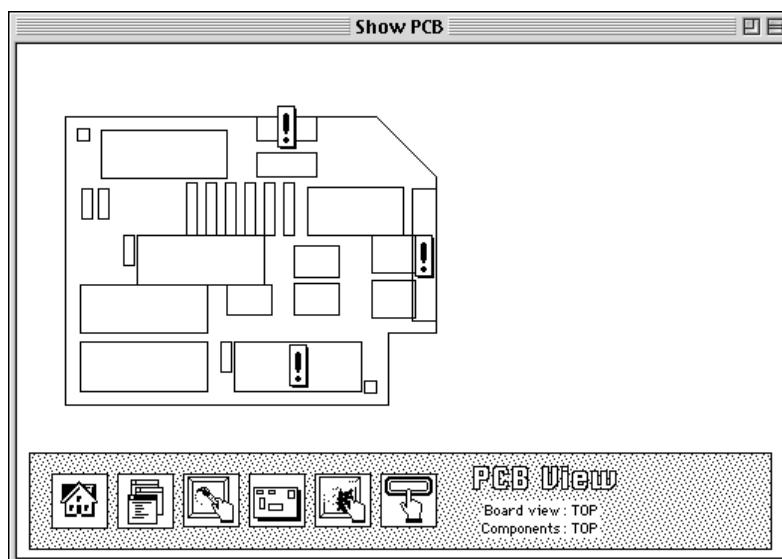


Figure 5.6: Prototype DIS screen identifying components in violation of a design guideline

Components which are in violation of one of the testability guidelines are shown overlaid by a button showing an icon like an exclamation mark. Clicking on this button takes the user straight to the corresponding page in the guidelines section of the DIS.

While this capability was never explored to any great degree, proof that it was possible to construct content 'on the fly' for graphical display within a multimedia system proved very valuable later. (This would be the basis for some of the programming work detailed in the next chapter). It was anticipated that screen displays generated to show the specific workpiece under consideration rather than some generic graphic would allow more effective communication with the designer.

5.6.2 DICTA Information System Flexibility

One significant problem in the creation of IEM software for electronics manufacture is the rate of change within the industry. To be competitive, manufacturers have to make use of smaller, lighter components and employ cheaper manufacturing processes. As processes and their parameters change, keeping IEM software up to date becomes difficult, due in part to the high development effort associated with software of this kind.

This problem is catered for within the DICTA environment to some extent; each testability rule has one or more parameters which can be varied to suit the manufacturing technology currently employed. With the parameters correctly set when a design is analysed for testability, the pages of the information system which are 'bookmarked' for the designer's attention during testability analysis will reflect the proposed design's suitability for the current capabilities of the business. Rules which are not relevant can easily be disabled, such as edge clearance rules when the PCB will not be clamped pneumatically (and therefore requires no space for gasketing).

5.6.3 Hyperlinks between information in the DIS

Rapid navigation through a large store of information to view the areas which are of interest can be achieved with hypermedia links as described in Appendix 1. The prototype DIS employed a hypermedia approach, allowing users to move rapidly between the subsets of information that they chose. This meant the DIS could also be of value in a conventional training situation; novices could be introduced to the principles of electronics fabrication and the equipment in use, treating the DIS as an electronic book. Explanations of unfamiliar terms could be found by following hyperlinks. By this stage the DIS contained many pages of supporting information, in addition to the rules pages themselves. Figure 5.7 shows examples of such glossarial pages; these do not refer

specifically to testability, but by addressing a wider array of manufacturing issues it was hoped that the testing process could be understood in context.

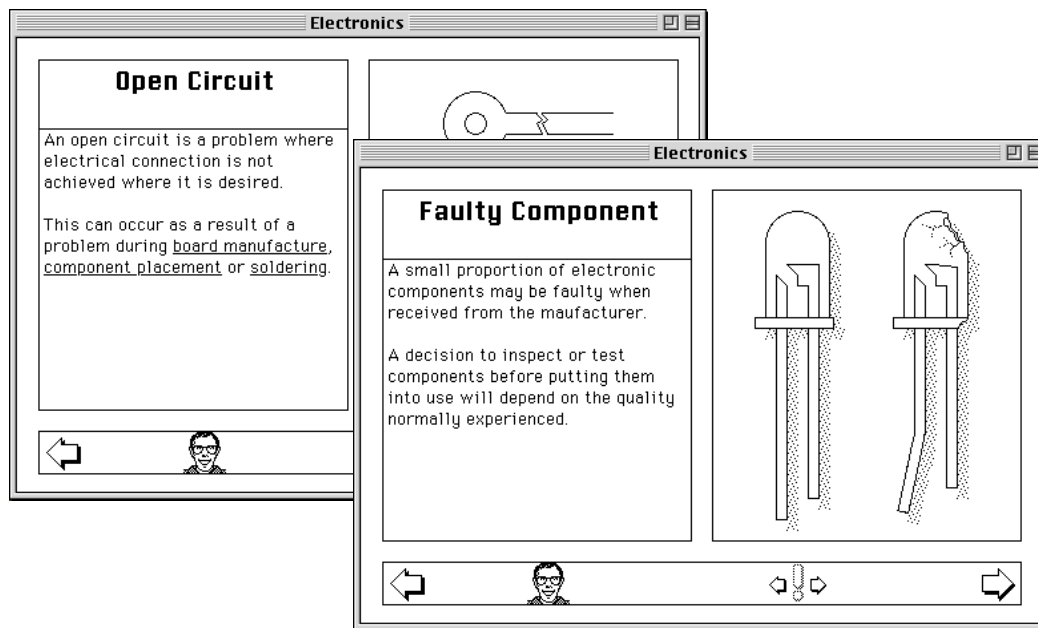


Figure 5.7: Supporting screens of glossarial information in the DIS

It was stated in Section 5.6.2 that changes occur rapidly in electronic product manufacture and information obsolescence is a real danger. The information held within a system like the DIS is likely to require ongoing modification as a result. Whole sections could become obsolete as production methods change, this being a characteristic of the electronics industry [Culverhouse, 1993]. Allowing selected users to have access to the software for the purpose of making changes to the content was not difficult, but two problems were clear:

- Editing content and creating additional pages for the multimedia information system were tasks for which skills might not exist within the business.
- As content changed, a conventional hypermedia system could be left with broken (invalid) links, or links pointing to out of date information.

As a possible solution to these problems, the DIS was modified to include a new mode of operation, somewhere between the read-only user mode and the full authoring mode employed in its original creation. This maintenance mode allowed the user to create additional blank pages, type text into fields and paste images into place, or edit existing information.

To address the problem of 'orphan' hyperlinks which were no longer useful as content changed, the field in each page was relied upon to hold information about the content of the page. When the user creates a page, the title and subheadings should hold any key words or phrases which describe the content of the page. When all maintenance has been done the system is returned to read-only mode.

To use the DIS, a user may select any word(s) on the page with the mouse driven cursor. The system will then search each page in an attempt to find a matching word or phrase in use as a title. If the users clicks with the mouse on any word within the main body of text, the system will move to the relevant page of information, if available. For example, a novice user who is unable to recognise the acronym 'THT' could click on it, and be taken to a page entitled "Through-hole technology (THT)".

Underlining was used to denote significant groups of words which might be of particular interest, and to make sure the user jumps to the right page. An example of an underlined word group might be 'test pad shape'. Selecting the individual words might take the user to other pages in the book if they were found first, such as pages entitled 'test points', 'pad orientation' and 'board shape'. With a group of underlined words, the system is forced to search until it finds a page that has the whole phrase in its title.

The dynamic method of linking between pages was not perfect since it was possible to highlight a word for which no matches existed. On failure to find the word nominated by the user, they are simply informed that no such reference exists. A future improvement might be the creation of a log of failed links so the person who maintains the information store can see which new pages might be required. The alternative remains to employ conventional 'hardwired' hypertext links made by a programmer. For a set of information subject to rapid change, this approach would require significant effort on an ongoing basis. The automatic generation of hypertext links was subsequently explored in the MIDAS project at the University of Salford (see [de Caluwe et al, 2000]).

5.7 The DICTA Information System under Multimedia ToolBook

While the HyperCard prototype had demonstrated that it was possible to undertake calculations within a multimedia package for the purpose of varying the content made available to suit the current situation, HyperCard was unsuited to the promotion of the DICTA guidelines in an industry that made very little use of Macintosh computers. A compromise involved creating a DIS package which would run under Microsoft Windows, which is common in UK industry. The 'Asymetrix Multimedia ToolBook' authoring language was found to be virtually a PC-based equivalent to HyperCard, and

therefore ideal. Both authoring languages organise information into pages and control the programme by a series of text-based scripts, typically associated with buttons. Virtually every command used in a HyperCard script had a ToolBook analogue, so it was clear that the authoring environment would support the requirements of a Windows version of the DIS.

ToolBook supported the display of quality colour graphics and colour buttons on icons which made the creation of an attractive package possible. In ToolBook, as in Microsoft PowerPoint, graphics were produced by drawing polygons, though bitmapped graphics in common formats could also be imported. For early prototyping work it was possible to cut existing images from both the PowerPoint slideshow and the HyperCard DIS and paste them straight into a sample ToolBook programme. In addition to diagrams, digitised photographs were added to relate concepts to the actual equipment and products more closely. Figure 5.8 shows an example.



Figure 5.8: Screen shot from the DIS under ToolBook

5.7.1 Programmed Animation as an Alternative to Digital Video

To offer effective guidelines it was necessary to show designers how test equipment moved, so they could see if their designs were likely to impede the movement of test probes. With Multimedia ToolBook, changes to the properties of polygons could be scripted, changing their location, the co-ordinates of individual vertices, line colour, line

thickness, and others. It was found that rapidly changing the properties of a polygon or group of polygons by small increments provided an attractive form of animation.

In multimedia computing, many displays which are referred to as animations are actually movie files – digitally stored as a sequence of images to be shown in rapid succession. Using the same technology as digital video, these are naturally no more or less efficient. They are called animations because the media shown is artificial, not a digitally presented film from real life. This is the moving picture equivalent of the difference between a photo and a diagram. Both have advantages; diagrams can be simplified representations for clarity and can show things which can't be photographed, while photos are more realistic.

Scripted ToolBook commands to move and reshape polygons allowed clear, simple animations to be created, showing how ICT equipment moved. Animations made in this way used only a fraction of the computer storage space necessary for conventional digital video. For example, the script extract given below is part of that which shows how a PCB is loaded to a test fixture and brought into contact with the test probes. Figure 5.9 shows the screen where the script is located. When the button 'demonstrate' is pressed, groups of polygons are moved or resized to produce a simplified animation of the operation of a pneumatic test fixture.

```
show irregularpolygon "arrow"
put "Air is pumped out of test fixture. Gaskets are compressed."
into text of
field "comments"
while i <=360
  move group "pcb" to 720,(1584+i)
  i=i+.25
end while

i=0
while i <= 216
  move group "top_plate" to 576, (2232+i)
  move group "gasket" to 648, (2020+i)
  move group "pcb" to 720,(1944+i)
  bounds of group "leftspring" = 576,(2376+i),792,2736
  bounds of group "rightspring" = 4536,(2376+i),4752,2736
  i=i+4
end while
```

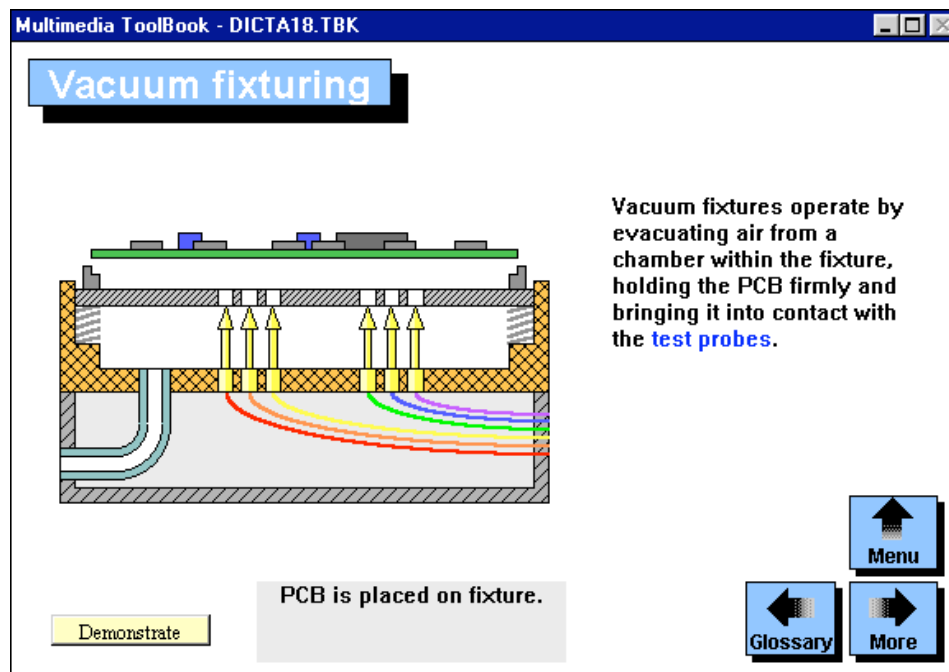



Figure 5.9: Vacuum fixturing principle as shown in the ToolBook DIS

The 'while ... end while' loops allow groups of polygons to be moved or resized as their coordinates have all been defined to include an 'i' component. Small increments to this variable achieve smooth movement over several seconds. The end result is an animation of approximately 20 seconds over about a quarter of the screen area. The script and the definition of the polygon groups which make up the image occupy a total of about 10K; a movie clip or 'animation' achieved as a movie, of the same screen area and duration, might require as much as 10Mb.

5.8 Conclusions

There is an often reported impression that multimedia fails to meet its expectations [Vanegas & Baker, 1994], and there appears to have been a focus on the technology itself rather than the development of applications which take advantage of its capabilities [Goble, 1994]. Hopefully, the DIS has demonstrated how multimedia might best add value in industry, allowing the end user the degree of control and customisation they have come to expect from other software such as a spreadsheet or database.

The dynamic hyperlink approach demonstrated in the DIS prototype successfully accommodated the addition of new information to the 'electronic book' as new pages. The users could simply add information as required, possibly to build up a collaborative picture of their working practices.

Development work undertaken to produce the DIS gave some insight into the pitfalls for the use of multimedia in manufacturing. Gathering information for presentation within the system was difficult, and when discussion with electronics industry staff did take place it revealed tremendous variety in the manufacturing equipment employed, parameters used and, consequently, which DICTA rules they considered important. As a result, flexibility within an ordered and maintainable programme structure was a key factor in making the DICTA information available in a useful multimedia form. It was necessary to accept that the 'customer' knew best, and offer a tool which could be configured and employed as necessary. It is concluded that multimedia will serve best in the electronics industry when it can be presented in the form of a functional shell to which company specific information can be added seamlessly, without recourse to complex programming.

Development effort was found to increase significantly when the information system ceased to be a simple linear slideshow and became something which could be browsed selectively, though this was largely offset in the HyperCard prototype when nominated hypertext links were abandoned in favour of a more flexible system. Careful planning of updates and periodic maintenance are still required, however.

The experimental work into the dynamic generation of multimedia content illustrated how multimedia software might be better able to pay back the costs inherent in its development. A linear multimedia programme would be exactly the same each time it was read, and might consequently be looked at only once or twice. Even an interactive store of information where users can read selectively will only be referred to occasionally. The DIS features three kinds of content which are altered dynamically in response to a current design situation:

- Text – lists of components in violation made available on each rule page
- Graphics – screen displays which show the location of components in violation.
- Buttons – on an 'overview' page, buttons were shown or hidden depending on the rules violated. Thus, the user's navigation options were also affected by current violations.

It is hoped that these features elevate the DIS beyond the passive 'electronic book' role where multimedia is often found. The up-to-date content which can be viewed in the DIS should make it a tool which it is worthwhile to consult regularly, rather than a simple training resource.

6.0 An IEM Tool to Support the Education of Novice Process Planners

In the previous chapter, an application was described where it was seen that interactivity could substantially increase the effectiveness of an educational message. Programming carried out showed that it was possible to do more than simply allow navigation within an existing multimedia document, improving the relevance of the lesson by varying the content to suit the user. Dynamic content generation (hereafter, DCG) had been proved to be possible within a multimedia environment, though it remained necessary to identify and quantify the benefits arising from the technique.

This chapter describes an application produced for that purpose, taking the form of an IEM programme to assist in the teaching of process planning to novice engineers. The subject is described, and the reader will see how teaching with conventional methods had been problematic. Software was created to facilitate better learning, also allowing the effectiveness of DCG to be tested.

6.1 A Definition of Process Planning

Process planning is the specification of the manufacturing steps required to make a new product, together with their key parameters. Considering the product design, production volumes and knowledge of the manufacturing resources available, the process planner (or 'methods engineer') forms a plan which defines the strategy for making the part, and its route through the factory. Various definitions of process planning have been made:

“Process planning consists of determining the sequence of individual manufacturing processes and operations needed to produce the parts in the job shop manufacturing system. The route sheet and operations sheet are the documents that specify the process sequence through the job. The route sheet lists manufacturing operations and associated machine tools for each workpart ... The operations sheet describes what machining or assembly operations are done to the parts at particular machines.”

[DeGarmo et al, 1990]

“Process planning may be defined as the function of transforming design information into work instructions. It covers the interpretation of drawings, the selection of raw materials in the correct form and shape (note the overlapping of this and the design and planning functions), the selection of the correct tools and

production methods and the sequence of operations. The objective of this function is to produce process plans that enable parts to be manufactured to the correct specification and in the most economical manner.”

[Wu, 1992]

Process planning can be considered to be one link in a chain between the activities of design and manufacture, where the intent of the designer is converted into practical instructions for volume manufacture (Figure 6.1). Chang and Wysk [1985] identified process planning as a vital integration stage between design and manufacture; clearly something of which student engineers should have an appreciation.

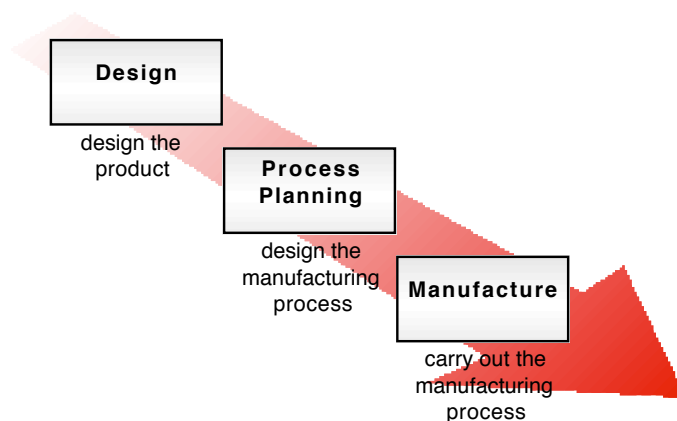


Figure 6.1: Process planning context

Effective process planning will provide valuable information such as the cost of making a product, the tooling which will be required and the impact that a new product will have on the capacity of the factory. In effect identifying requirements clearly helps to ensure that a design is viable before manufacture is attempted.

6.1.1 Contents of the Process Plan

As stated in the definition of process planning given by DeGarmo et al [1990], a process plan typically takes the form of a route sheet and an operations sheet. The route sheet identifies both the processes and the machines to be used. An example of a route sheet is shown in Figure 6.2.

These documents have a heading identifying the product, and a body section comprised of numbered instructions. Work instructions are in a hierarchic format; for each of the stations listed on the route sheet, a more detailed set of instructions exists in the form of an operations sheet, a sample of which is shown in Figure 6.3. Each operations sheet is a set of detailed instructions for the operations to be carried out on a batch at one

machine or cell before the whole batch advances to the next setup. Parameters such as the cutting speed and feed rate for each machining operation are included.

Salford Engineering		Route Sheet	
Part Number	HSPA1	Planner	R. Farr
Part Name	Retaining Spindle	Version (date)	2 (29/06/92)
Quantity	1,000	Material	SAE 1040
OPERATION NUMBER	FUNCTION	EQUIPMENT OR MACHINES	TOOLING
1000	Issue bar stock from stores		
2000	Saw into 98mm billets	Billet saw	
3000	Turn to drawing (HSPA1-1.2)	C&L Turret lathe	#541 box tool
4000	Heat treatment, 1700°F for 30 mins,	Atmosphere furnace	
5000	Inspection	Coordinate measuring machine.	Fixture #V04

Figure 6.2: Sample route sheet

Salford Engineering		Operations Sheet		
Part Number	HSPA1	Planner	R. Farr	
Part Name	Retaining Spindle	Version (date)	2 (29/06/92)	
Quantity	1,000	Material	SAE 1040	
OPERATION NUMBER	FUNCTION	TOOLING	PARAMETERS	
			SPEED	FEED
3000	Receive 1000 off 98mm sawn	C&L Turret lathe		
3010	Grip in chuck, extension =	3 jaw chuck		
3020	Face off, depth = 1mm	RH facing tool	40	0.01
3030	Centre drill, depth = 6mm	Combination centre	750 rpm	hand
3040	Straight turn, X=0 L=40 D=48	RH round nose tool	40	0.008
3050	Straight turn, X=0 L=20 D=32	RH, small radius	40	0.005

Figure 6.3: Sample operations sheet

In some businesses the process plan will be little more than a single piece of paper that travels with the work to identify it. This is acceptable under some circumstances, such as where product variance is very high (a highly skilled workforce is employed to react as required) or where product variance is non-existent (operations need not be specified because processes are standardised to a great degree).

6.2 Teaching of Process Planning at the University of Salford

At the University of Salford, process planning is taught to undergraduates studying Manufacturing Engineering and Manufacturing Management, and also to some taught course M.Sc. students. Initially, teaching of process planning was carried out with a lecture on the principles of process planning, followed by a practical session in which students would work in groups to interpret and make changes to an existing process plan. Students were expected to produce a new version of the plan, stage drawings (diagrams showing the current shape of the workpiece, methods of clamping and the tooling in use), and a short write-up justifying the planning decisions taken.

Some students completed this assignment with ease but it was clear that other students had no experience of working with machine tools. Some plans would have produced misshapen parts, and others included nonsensical instructions, indicating a student whose understanding of the equipment was limited. Trainor and Dore [1994] found that the backgrounds of students on these courses varied tremendously.

Some students had specified machining operations which might have damaged equipment or placed its operator in danger. These students needed experience, but they could not be allowed unsupervised access to machine tools. It was apparent that a gap between theoretical knowledge and practical application remained unbridged. Demonstrations were not ideal as group numbers were often too large for everyone to see properly, and the pace of such a demonstration may not have suited all the students. It was thought that IEM software might offer a solution to the limitations of the conventional training system.

It was aimed to discover how IEM software might best be used to improve the teaching of manufacturing process planning. First, a prototype was created in HyperCard, as a feasibility study. Later this was superseded by a programme created with Multimedia ToolBook. In each case, experience gained in DICTA project software development (detailed in Chapter 5) was valuable.

6.2.1 A Sample Exercise in Process Planning

In order to give students an appreciation of process planning, a laboratory exercise was conducted which placed groups of students in the role of process planner. At first, no use of computers was made.

The student exercise concerned a simple rotational part. Students worked in small groups, studying a part drawing and listing the manufacturing instructions which would allow a product of the same shape to be turned from a solid bar. The material to be removed was identified by giving values to the parameters X , L and D for each cut. This format is employed in Zhang and Altin [1994], a textbook from which the students had been given examples. Figure 6.4 illustrates this; X is the distance down the bar from a datum before a feature begins, L is the length of the cut made, and D is the final diameter to which the workpiece is reduced. With a few extensions such as the addition of parameters for depth of cut, feed rate and cutting speed, a set of instructions in this format can unambiguously specify the manufacture of any turned part – once the student has the knack of breaking down a design into a series of machining operations.

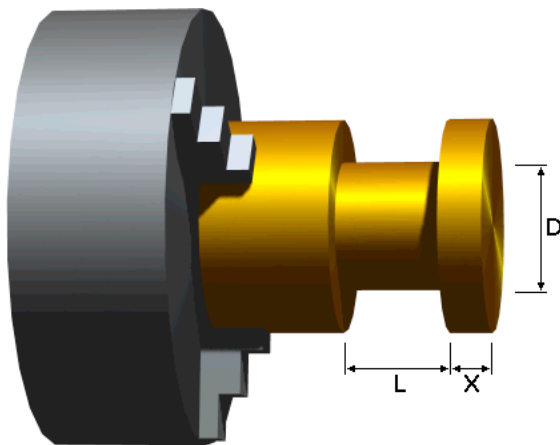


Figure 6.4: Basic protocol for specifying turning operations

Under conventional instruction, during the lab itself and when marking assignments, it was necessary for the tutor to interpret students' plans, in each case describing what the result of their instructions would be. This typically involved sketching a stage drawing quickly to show the nature of any problem. An example follows; in this simplified, fictional exercise an operations sheet is required which will allow a machine operator to produce collars like the part shown in figure 6.5:

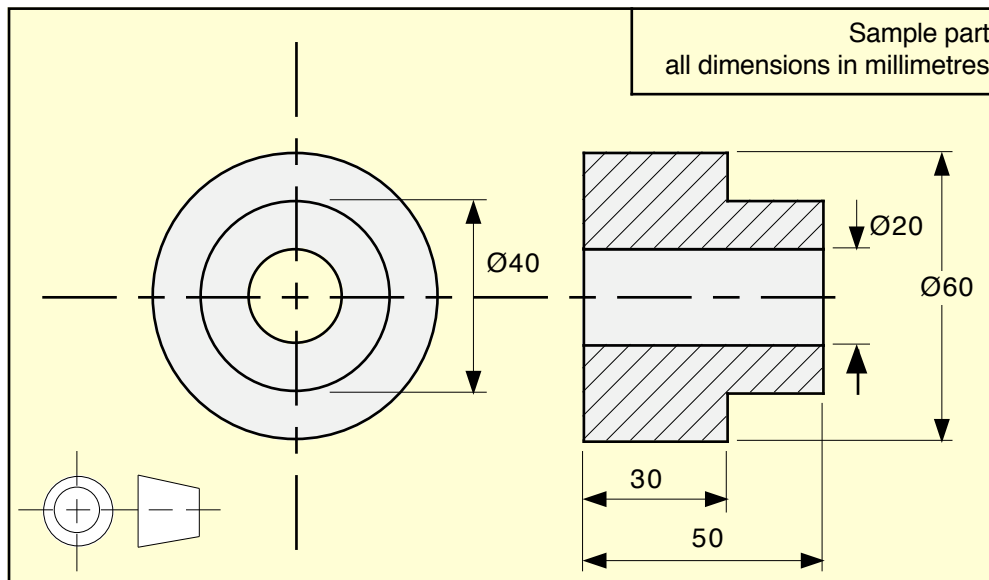


Figure 6.5: Simple part requiring an operations sheet

A group of students attempting to produce an operations sheet for the manufacture of this part might generate a set of instructions like those given below:

```

2000  Receive from stores, aluminium bar stock Ø65mm
2010  Grip bar, extension = 30
2020  Straight turn X = 0, L = 50, D = 60
2030  Straight turn X = 0, L = 20, D = 40
2040  Centre drill, depth = 8
2050  Drill, depth = 50, diameter = 20
2060  Part off, X = 50

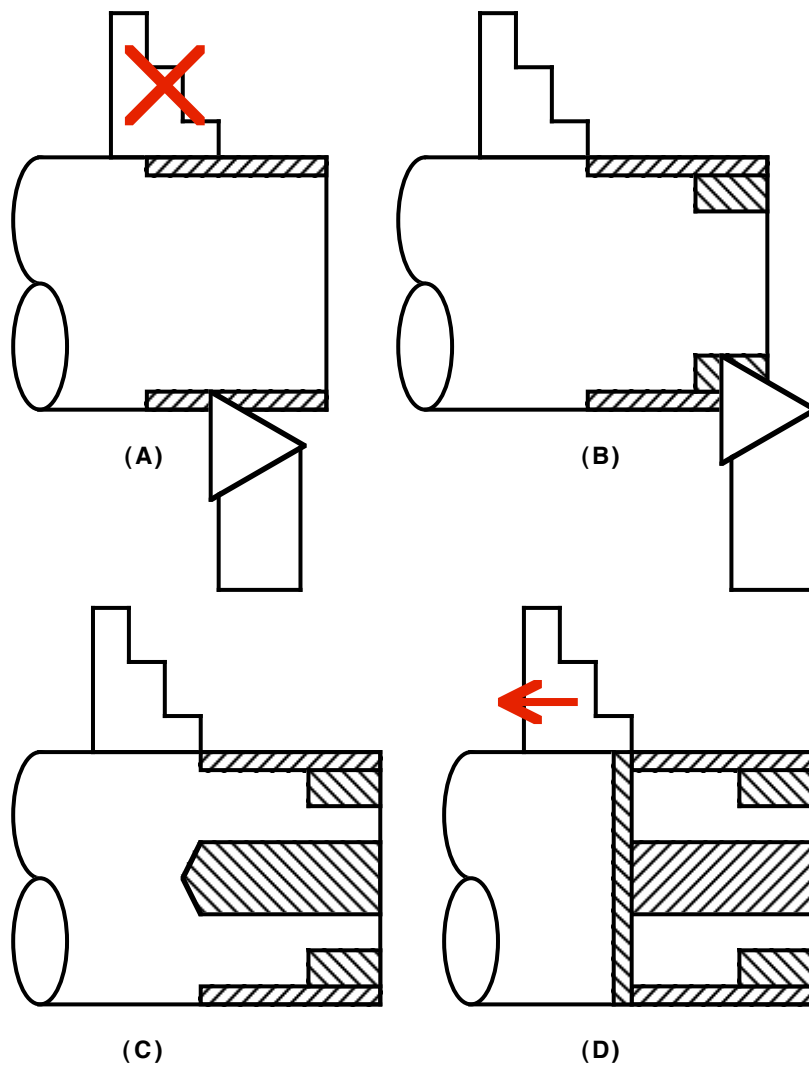
```

Once the students write a process plan it is necessary for a tutor to interpret it. This phase of the exercise is the reinforcer, confirming that the learners have done everything right, or demonstrating where problems lie. The feedback that students might receive for the operations list given above follows. It is verbal, but also involves sketching some simple diagrams, shown in Figures 6.6 A–D (page 76).

“Okay, so we’re gripping the bar with an extension of 30...” the tutor starts the diagram with a picture of the bar sticking out of the chuck. “But then we straight turn, X=0, L=50, D=60.” The tutor adds a hatched area, showing where material is to be removed. The instruction the students have given requires the operator to machine away part of the surface of the bar where it is clamped in the lathe chuck. This is impossible; attempting it will result in a collision between the chuck and the tool. The tool will probably snap if these instructions are followed. The tutor adds a picture of the tool being

used (Figure 6.6A). “What happens as the tool moves down the bar?” The students see the problem. If they want to machine 50mm down the outside of the bar, they have to grip the bar with a greater extension. “Now, Straight turn $X = 0$ $L = 20$ $D = 40$. What area of material does that remove?” A student correctly identifies the ‘step’ feature on the outside of the bar and it is added to the diagram (Figure 6.6B). The centre drilling and drilling operations pose no problems, so they are also added as hatched areas (Figure 6.6C). “Finally, part off, $X=60$. And that does what?” A student adds the last hatched area (Figure 6.6D). “Okay, again you’re at risk of a collision with the chuck, but once you change the extension value in line 1010, you should be all right. One last thing: you never faced off the bar. What if the bar material came to you with a rough end?”

Thus, the students are required to rework their operations sheet briefly, before moving on to a more complex exercise. Though the example presented is fictional, the question and answer style to the feedback process is typical of what was carried out in the classroom. It was observed that the practical element of the students’ tuition in process planning involved a great deal of input from the tutor. In requiring the students to act as process planners rather than relying solely on a formal lecture, the learners’ understanding of the subject was increased, but there were practical limitations to this method of delivery. Each group of students periodically required the attention of the tutor, who had to interpret their list of machining operations correctly and explain any problems which arose, leaving the students to make improvements as necessary. The tutor performed a similar but not identical task with each group before moving on. Essentially, the division of the tutor’s time limited the amount of assistance which could be given to struggling students, and also prevented the exploration of more complex issues by students who found the exercise easy.



Figures 6.6 A–D: Sample stage drawings

Teaching the principles of process planning in this way was hard work, even with the assistance of a second tutor in a class of 25 students, and some students were still producing process planning assignments of poor quality. Practical exercises in process planning were considered to be an ideal subject area in which to prove the value of IEM software with a DCG element. Advantages were anticipated:

- Since the tutor’s feedback was a combination of illustration and commentary, multimedia would be necessary to achieve the same communication with the student.
- Feedback needed to be given as rapidly as possible, to act best as a reinforcer – CAL gives each student (or group at a computer) their own feedback.

-
- The tutor's input was essentially repetitive in nature, consisting of a number of simple checks. It would therefore be relatively easy to capture this knowledge for presentation on the computer.
 - Animations would improve the quality of feedback, giving a realistic impression of how the machine tools operate.

6.3 An IEM Training Tool for Process Planning – The HyperCard Prototype

A prototype piece of software was created to test the concept of representing a machining operation by DCG animations. Programming began using HyperCard, taking advantage of the knowledge which had already been gained in the generation of schematics of populated circuit boards for the DICTA Information System (detailed in Chapter 5).

With HyperCard it was found that it was possible to control the operation of the software environment using scripted commands which controlled the user interface as if it were being manipulated by a user. Thus, painting tools could be selected and moved about the screen to build up images in runtime. For example:

```
lock screen
set grid to false
set filled to true
set pattern to 1
choose rectangle tool
drag from trunc(100 + extn),100 to trunc(100 + extn - matlen),
trunc(100 + matd)
choose browse tool
unlock screen
```

This script fragment illustrates the basic principle which was used to achieve DCG under HyperCard. First, the user's control over the graphic user interface is suspended with the command 'lock screen'. Normally, the user is able to use the mouse and keyboard to make selections as they work with the software, but in order to draw items direct onto the screen, control of the cursor is subverted to the script itself. The next command, 'set grid to false' ensures that any shapes drawn on screen will not appear misshapen as a result of their vertices snapping to the nearest points on a grid which can be invoked when drawing manually. This is undesirable for DCG images because it effectively reduces the display resolution achievable. 'Set filled ...' and 'set pattern ...' ensure that the

shapes to be drawn will appear in the right style. 'Choose rectangle tool' selects the tool with which boxes are drawn on the screen. The command 'drag from ... to ...' acts exactly as if the user had moved the mouse. The cursor (hidden by the lock screen command but still active) moves from point (x1,y1) to point (x2,y2) defining a rectangle by its opposite corners. The 'trunc' function forces values to be integers, since 'drag' will not work with floating point numbers. The rectangle appears, its size and location determined by the variables matd (bar diameter), matlen (bar length), and extn (extension beyond chuck). Finally, 'choose browse tool' and 'unlock screen' return the user to the normal operating mode. The user would then see a screen display something like the one in Figure 6.7. Because the rectangle representing the bar is positioned and sized according to variables, the user sees a diagram which matches the values they have specified in their process plan; dynamically generated content.

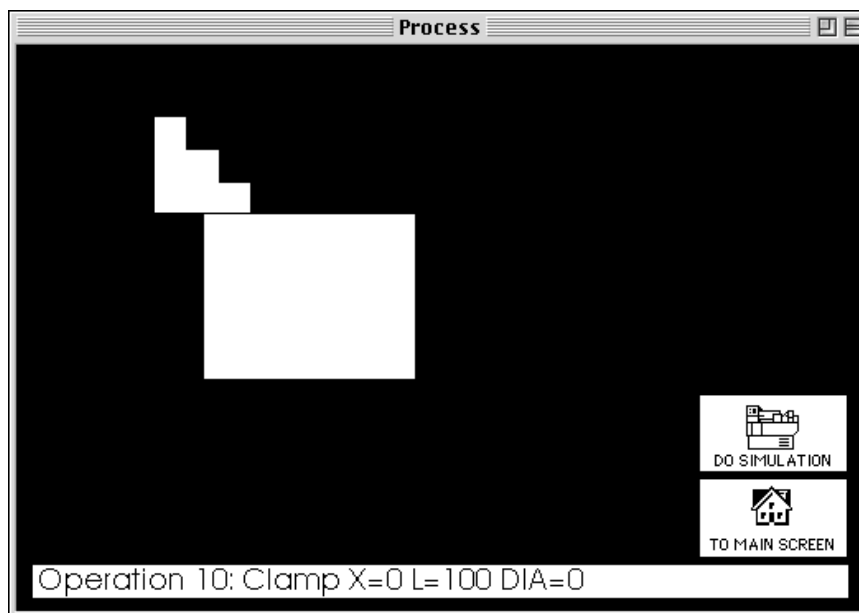


Figure 6.7: Bar material clamped in the chuck, HyperCard prototype

Using the same scripting technique, it was possible to draw other rectangles, all based on the user's process plan. Using 'set pattern to 12' meant any rectangles drawn were black, this representing areas where material had been removed. Line by line, the user's process plan was interpreted and the screen updated to show the geometry of the workpiece by blacking out areas as they were machined away.

It was a simple matter to add tests for common student errors. For example, when a cutting operation would bring a tool into contact with the chuck:

```
if (extn - x - 1) < 1 then
  beep 1
  unlock screen
  answer "Oops - just machined a lump out of the chuck!" with "Stop"
  choose browse tool
  exit mouseup
end if
```

Thus, any problems in the student's process plan bring the simulation to a premature end, with a message describing the problem. Because every group of students has a computer critiquing their process plan, feedback is more rapid and therefore a more effective reinforcer.

It was found that in addition to simply drawing shapes on screen, parts of an image could be selected and then dragged around, supplying an illusion of motion. In this way, the animated machining sequences were made more realistic. The fragment of script shown below uses the 'lasso' tool to copy an existing image from a database of different tools. It is then pasted into the simulation page and moved about using the same 'drag' command which was employed earlier to draw the rectangle representing the bar of material.

```
if opType = "Face off" then
  lock screen
  go to card 17 of background "Tools"
  choose lasso tool
  doMenu "Select"
  doMenu "Copy Picture"
  go to card 1 of background "Simul"
  unlock screen
  doMenu "Paste Picture"
  set dragSpeed to 40
  drag from 319,240 to trunc(100 + matl - 1),trunc(100 + matd + 1)
  drag from trunc(100 + matl - 1),trunc(100 + matd + 1) to trunc(100
    + matl - 1), trunc(100 + (matd / 2)) with optionKey, commandKey
  drag from trunc(100 + matl - 1),trunc(100 + (matd / 2)) to 319,240
    with optionKey, commandKey
  set dragSpeed to 0
end if
```

In HyperCard, dragging a graphic with the command and option keys held down leaves copies of the graphic behind as it is moved. By adding the command 'with optionKey, commandKey' to a script, this also happens. When dragging the tool graphic

only slowly ('set dragSpeed to 40'), it constantly leaves a trail behind (made by its own black border) as it is moved (see Figure 6.8). The simulation thus appears to show material being removed from the bar.

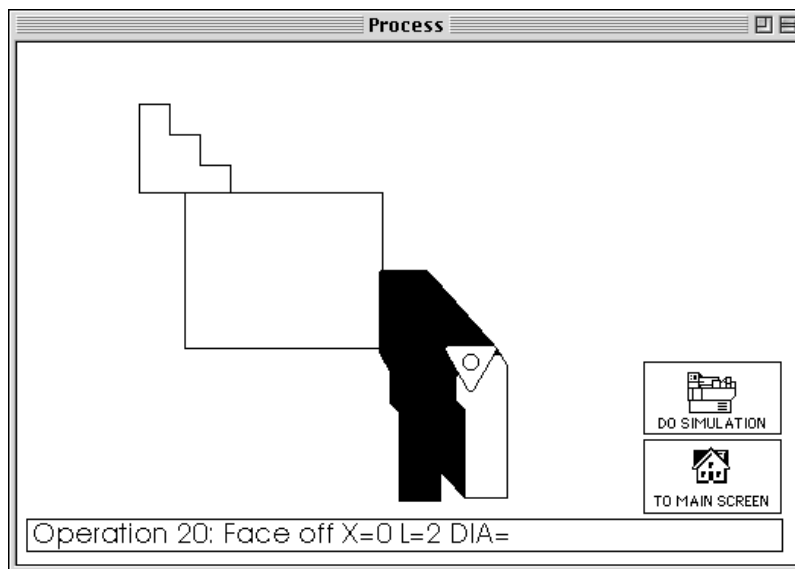


Figure 6.8: Animation of a tool path, showing how the tool graphic obliterates any 'material' it touches. (This is normally done against a black background.)

Figure 6.8 shows a tool tipped with a removable tungsten carbide insert in use. An earlier version of the IEM software had made use of the catalogue of tools shown in Figure 6.9, all of which are conventional one-piece tools which would be ground for sharpening. Although this was functional, consultation with experienced machinists resulted in a revision of the tools library; most manufacturing businesses now use disposable carbide inserts because it is not necessary to compensate for changing tool length as when conventional tools are sharpened. It was naturally desirable to have the software represent real life as much as possible.

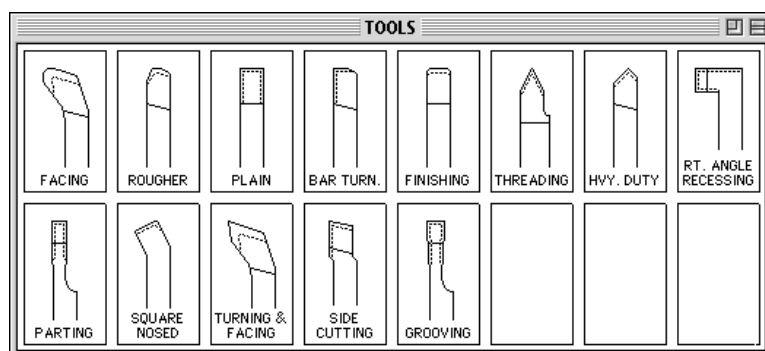


Figure 6.9: Tool library from an early version of the process planning software

One unfortunate limitation of an animated approach where tools are dragged into contact with the representation of the workpiece is that it is not easy for the software to do anything else at the same time. Thus, it is not possible to show material being removed from both sides of the bar simultaneously. Releasing control the lassoed graphic (tool) would be necessary each time the other side of the bar was updated, which would make the animation unreasonably slow. Instead, a tool is moved through its complete path in accordance with a single line of the student's process plan. The display is updated to bring both sides of the bar into conformity once the operation is completed:

```
choose select tool
drag from 0, trunc(100 + matd + 1) to trunc(105 + matl), trunc(100 +
                                                                    matd / 2)

doMenu "Copy Picture"
lock screen
doMenu "Paste Picture"
drag from 0, trunc(100 + matd + 1) to 0, trunc(100 + matd / 2)
doMenu "Flip Vertical"
unlock screen
```

This script fragment shows how the upper side of the bar shown in the animation is made to conform to the shape of lower side once a tool path animation ends. Basically, the screen area showing the lower half of the bar is selected, copied, pasted, dragged into place and mirrored vertically, all with visible screen updates suspended. Strategic use of the 'lock screen' command restricted the amount seen by a user to that which was relevant to the machining operation represented.

6.3.1 Results with the HyperCard Prototype

Once it was clear that the prototype could interpret a user's process plan and supply an animated output, it was possible to make use of the system with students. Even in its most basic form, the 'interactive process planning tutorial' was potentially superior to human instruction in some ways; it provided a very rapid form of feedback, and students could learn about turning experimentally without placing themselves in danger (or using expensive bar material).

The first user trial of the process planning software took place in December 1995, when twelve students on an M.Sc. taught course were given an hour to work through a simple exercise in groups of two or three. Their process planning task is shown in Figure 6.10. At this stage the software only supported a few machining operations. Most notably,

there was no facility for taper turning, nor for turning a workpiece round to form features at both ends. The part geometry required in early exercises was consequently simplistic. Still, the manufacture of the part in this exercise required that the user order the right material from stores, specify how to grip it suitably and then carry out facing off, centre drilling, drilling, bar turning and parting off processes. In choosing the operations required and their sequence, and detailing them unambiguously, the students developed an appreciation of the work of the process planner.

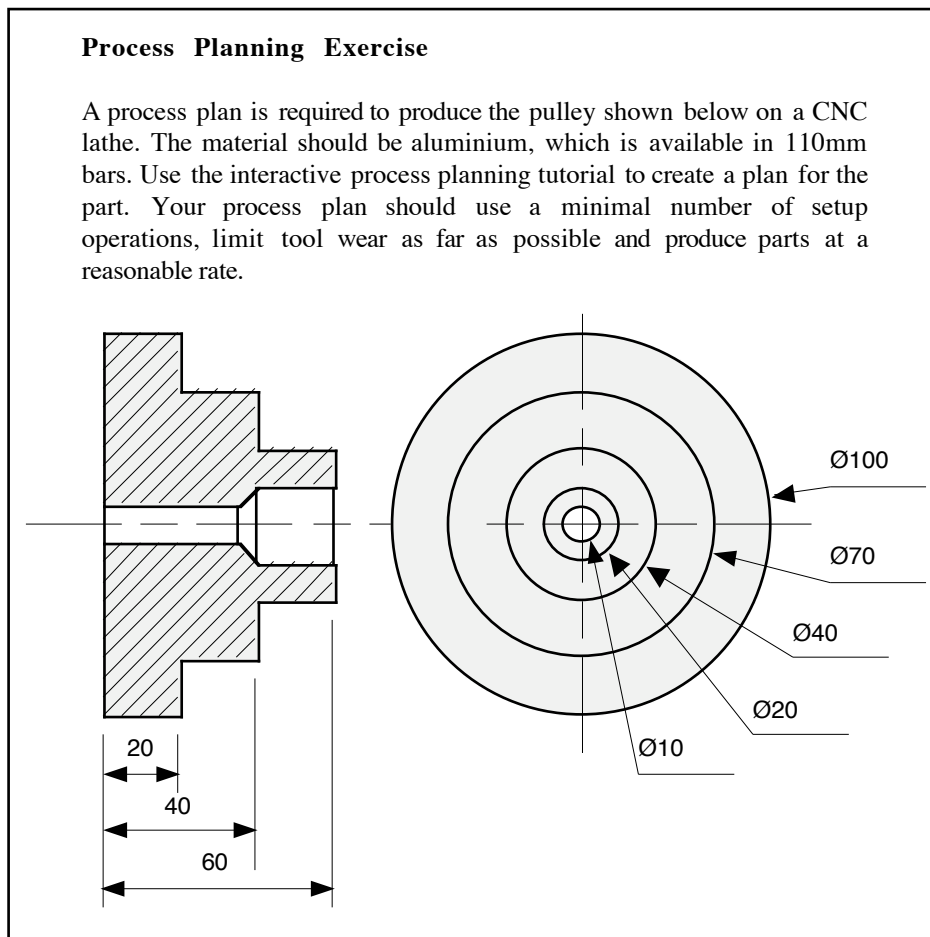


Figure 6.10: The first student process planning exercise

Assistance was kept to a minimum, with students encouraged to experiment with the program and explore the on-line documentation. The use made of on-line documentation illustrated how distinctly different approaches could be taken with the software. One pair of students made particularly great use of the help facility. Finding this by following a hyperlink to a specific page when they encountered an unfamiliar term, they then went back to page one of the 'book' and read through all the information provided, page by page (Figure 6.11 shows an example of the information available in the

manual). This approach was in sharp contrast to another group where the technique was much more experimental: “Simulate that – let’s have a look at it.” The experimental technique resulted in the exercise being completed in some 25–30 minutes; considerably faster than by those who attempted to guarantee success through careful planning on paper. Still, both rigorous learners and experimenters were free to operate in the way with which they were most comfortable. Of the five groups (three pairs and two threes) who did this process planning exercise, four had completed process plans which produced parts of satisfactory geometry, and the remaining plan lacked just the final instruction, parting off.

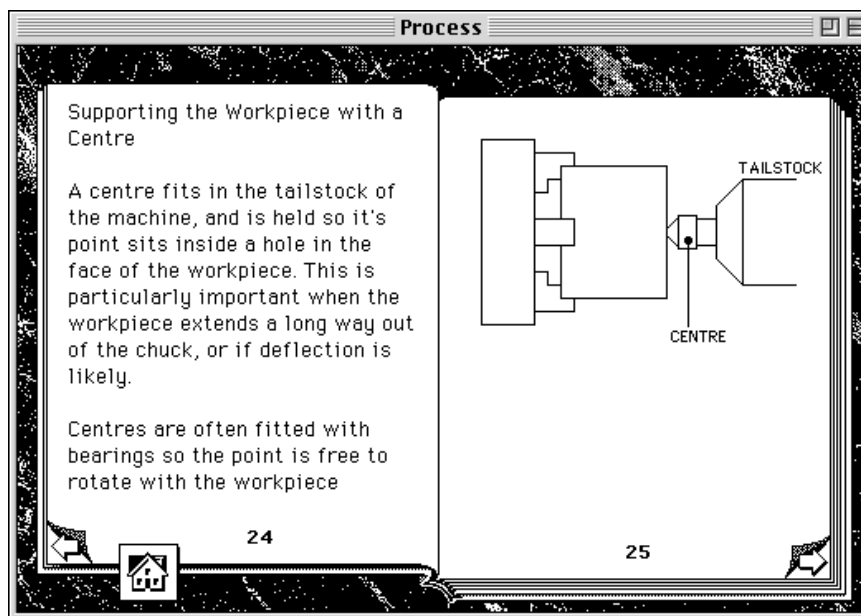


Figure 6.11: On-line documentation in the HyperCard prototype

Results were encouraging; a clear improvement in the students’ understanding of the subject could be seen towards the end of the exercise, when appropriate machining processes were selected and instructions entered without error [Farr and Lawlor-Wright, 1996]. The usefulness of the on-line documentation had been demonstrated, and in a one hour computer-based exercise students were producing process plans which were more accurate than had previously been achieved in a conventional lab exercise over three hours.

6.3.2 Student Responses to the Prototype

Having completed the exercise, each student was requested to complete an evaluation form. This helped to assess the users’ reactions formally, though students’

comments had also been noted during the exercise. First came a group of five questions (Figure 6.12):

Consider the following statements, and circle one number where 1 = strongly disagree, and 5 = strongly agree.					
1. The built-in help function was useful.	1	2	3	4	5
2. The exercise we were required to do was too easy.	1	2	3	4	5
3. The way the program operated was clear.	1	2	3	4	5
4. It was easy to understand how to define a process plan.	1	2	3	4	5
5. The program adequately supports the theory we had learned.	1	2	3	4	5

Figure 6.12: First section of users' assessment form, employed with the prototype

All twelve students returned filled out forms, giving responses for these questions as follows. Question 1, "The built-in help function was useful" received a mean mark of 4.25, which appears to suggest that the on-line documentation was suitably constructed and easy enough to access.

Question 2, "The exercise we were required to do was too easy" received a mean mark of 3.83 – ideally, a mean mark of 3 would have resulted, signifying the the exercise was generally considered to have been pitched at the right level. Clearly, future exercises would have to feature more complex parts requiring a wider variety of manufacturing operations. On a positive note, it may be that students were enjoying using the software, and considered it 'too easy' when they rapidly completed the exercise.

Question 3, "The way the program operated was clear" received a disappointing mark of 3.83. Some students had found it difficult to enter the lines of manufacturing instructions which make up an operations sheet. The interface of the prototype used a single screen to specify all machining operations, with multiple pop-up menus and other buttons, as shown in Figure 6.13. Though this allowed complex process plans to be entered, it presented a daunting variety of options when first encountered. Even so, all groups were operating the system successfully by the end of the exercise. This interface was improved, and finally replaced by a multi-screen system which more closely resembled a conventional paper-based process plan, to reduce information overload, described in Section 6.4.3.

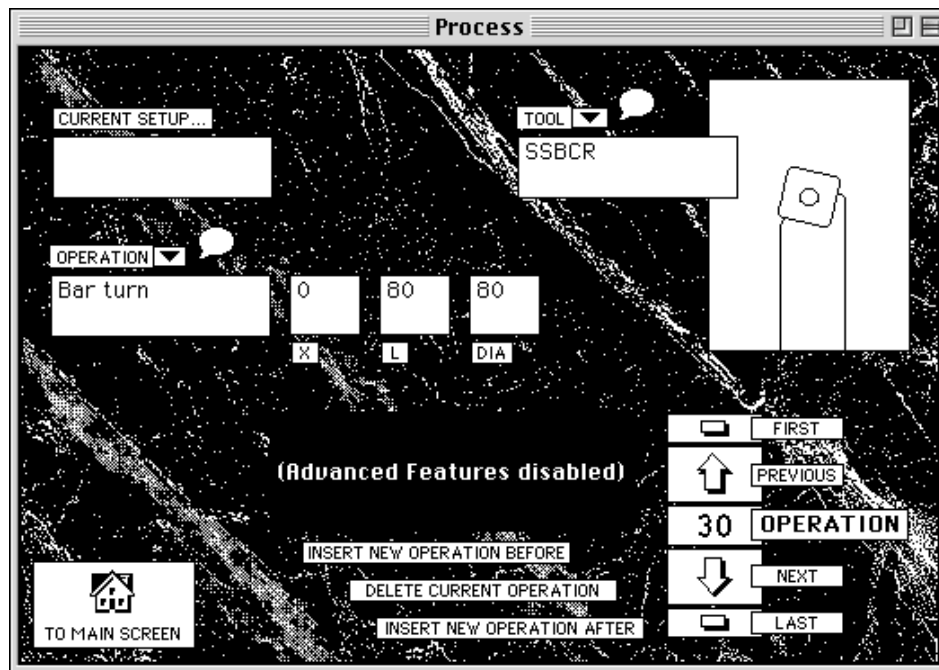


Figure 6.13: Entering a process plan – the prototype’s user interface

Question 4, “It was easy to understand how to define a process plan” received a mean mark of 4.16, which was encouraging. By the end of the exercise, all the students understood how to think of a part as a number of features, each to be achieved by operations in sequence. Students’ accuracy in specifying turning operations had been seen to improve during their time on the computers, and this mark would appear to reflect their confidence.

Question 5, “The program adequately supports the theory we had learned” received a mean mark of 4.33, another good result. It seems that using the program was considered a worthwhile exercise.

A second section of the student evaluation form featured two simple questions (Figure 6.14):

Do you feel that interactive tutorials of this nature are more enjoyable than conventional practical exercises?	(delete as appropriate) Yes / No
Do you feel that doing the interactive tutorial was more worthwhile than a conventional practical exercise?	(delete as appropriate) Yes / No

Figure 6.14: Second section of users’ assessment form, employed with the prototype

For the first question, all twelve students answered 'yes'. While it is not the sole purpose of an educational exercise to be entertaining, it helps considerably in keeping the attention of the user, so for a piece of IEM software to be considered 'enjoyable' is worthwhile. For the second question, ten students answered 'yes', and one of the students who answered 'no' to this question added a comment that performing a process planning exercise both ways (i.e. conventionally and with IEM) would be most worthwhile.

These figures confirm that in general the students found the activity stimulating. They were also invited to add comments on the best and worst aspects of the software, and to identify any other features they would have liked to see. Students liked the way pop-up menus eliminated typing, considered the software to be 'good visually' and found the animations valuable in checking their process plans. Negative comments related almost exclusively to the interface for entering process plans. One student expressed a dislike for working with Macintosh computers, and one reported a fault whereby a piece of bar material could be 'gripped' with extension greater than its length. This produced a strange situation where a bar would 'hang' in mid-air during machining. It was simple to build a check for this into the software. The menu for tool selection was identified as both the best and worst feature of the software, by different students, which shows how difficult interface design for IEM software can be. One student suggested that the program would be better if colour displays were used throughout. This was not possible within early versions of HyperCard, but full use of colour was made when the software was rewritten in Multimedia ToolBook. Finally, one student commented that it was the first time he had been able to use a lathe, albeit a 'virtual' one. Clearly, it must always be borne in mind that IEM software may be employed by students from a variety of backgrounds. What is familiar to one user is remedial to another, and entirely new to a third.

Around this time the IEM software for process planning acquired the name 'CAL3P', an acronym for 'Computer Aided Learning of Process Planning Principles' which was kept for the HyperCard prototype and the forthcoming Multimedia ToolBook version.

6.4 An IEM Training Tool for Process Planning – with Multimedia ToolBook

Development of the interactive tutorial continued, with the most significant single alteration being a change in authoring software used. CAL3P was reprogrammed to run under Multimedia ToolBook. At the University of Salford and in the UK in general, Macintosh computers were found to be quite rare, Windows compatible machines having been adopted as the standard. It was much easier to make use of IEM software that operated under Windows or MS-Dos.

With knowledge gained from student comments and demonstrations to people in industry, the Multimedia ToolBook version of CAL3P acquired several key differences from the earlier prototype, detailed in the sections which follow.

6.4.1 Colour Graphics

The Multimedia ToolBook authoring environment allowed full use to be made of colour. HyperCard originated in 1987 when few Macintosh computers had a colour capability, and even modern versions of the software make only poor use of colour. Although digital images such as scanned photographs could be shown in colour, script commands for using colour were inadequate; DCG images could not be multi-coloured.

Advantage was taken of the new capabilities which Multimedia ToolBook offered CAL3P; photographs were added to the software to provide an attractive introductory screen and to show real machine tools in addition to simplified graphics. Figure 6.15 shows the CAL3P main menu, making use of a scanned photo and colour icons. It was hoped that features such as this would give the software immediate appeal.

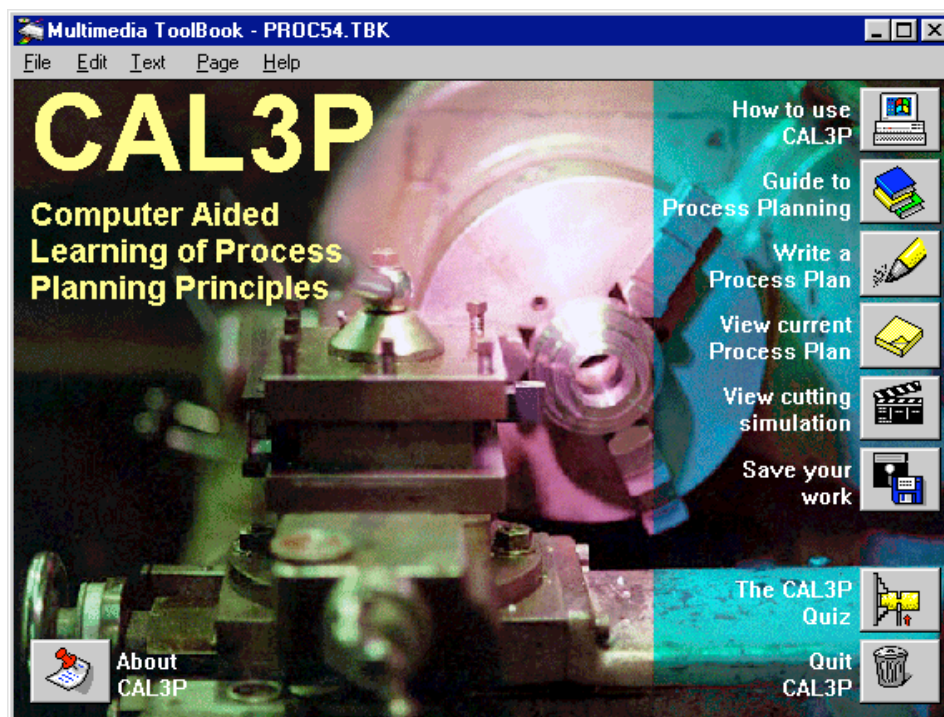


Figure 6.15: CAL3P main menu under ToolBook

6.4.2 Changes to the User Interface

In response to students' reports of difficulties with the interface when using the HyperCard prototype, the new version of CAL3P featured a simpler interface for entering a process plan. The operations sheet is presented as a form which the user is required to

fill in over several stages, each with a screen in the software. Figure 6.16 shows one of these. It was hoped that this approach would reduce the business of creating a process plan to a series of logical steps, still leaving the engineering decisions to be made, but not requiring the learner to puzzle over the operation of the software itself.

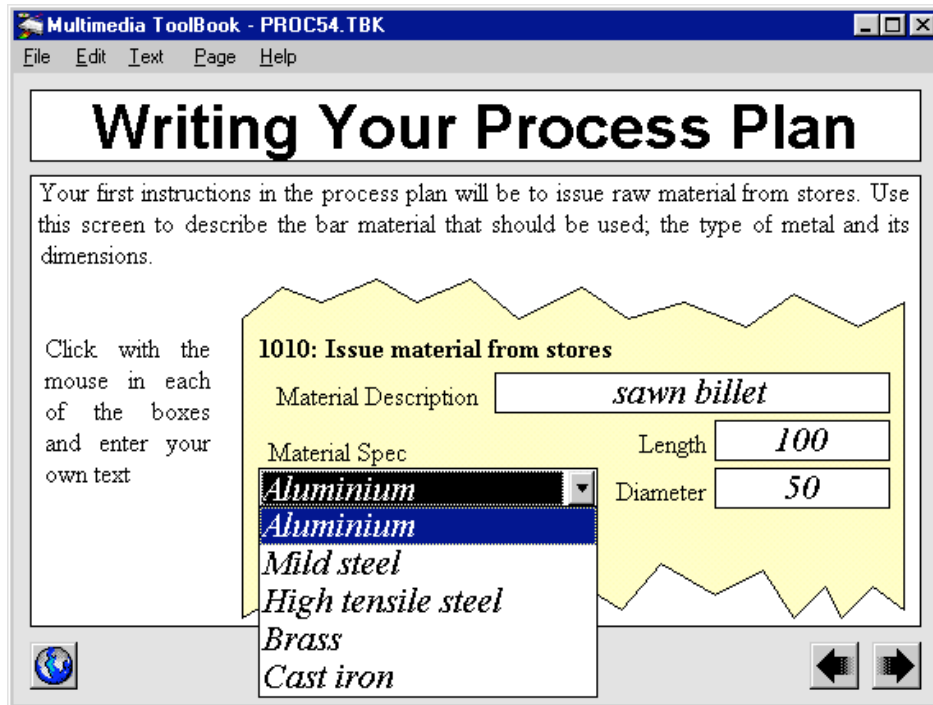


Figure 6.16: 'Writing Your Process Plan', specifying the material which will be used

The form paradigm is continued when the student moves on to specify individual manufacturing instructions. These are entered one line at a time, against an operation number. The screen for data entry does not attempt to show the whole plan at once, but only the instruction which can currently be edited, the previous line and the next line (Figure 6.17).

The 'up' and 'down' buttons allow the student to move from line to line. Changes to the central line can be made by the methods which would be employed in a typical word processing package; typing the instruction in directly, using the mouse to highlight sections for overtyping, using the cursor keys and the backspace key, or using cut and paste functions.

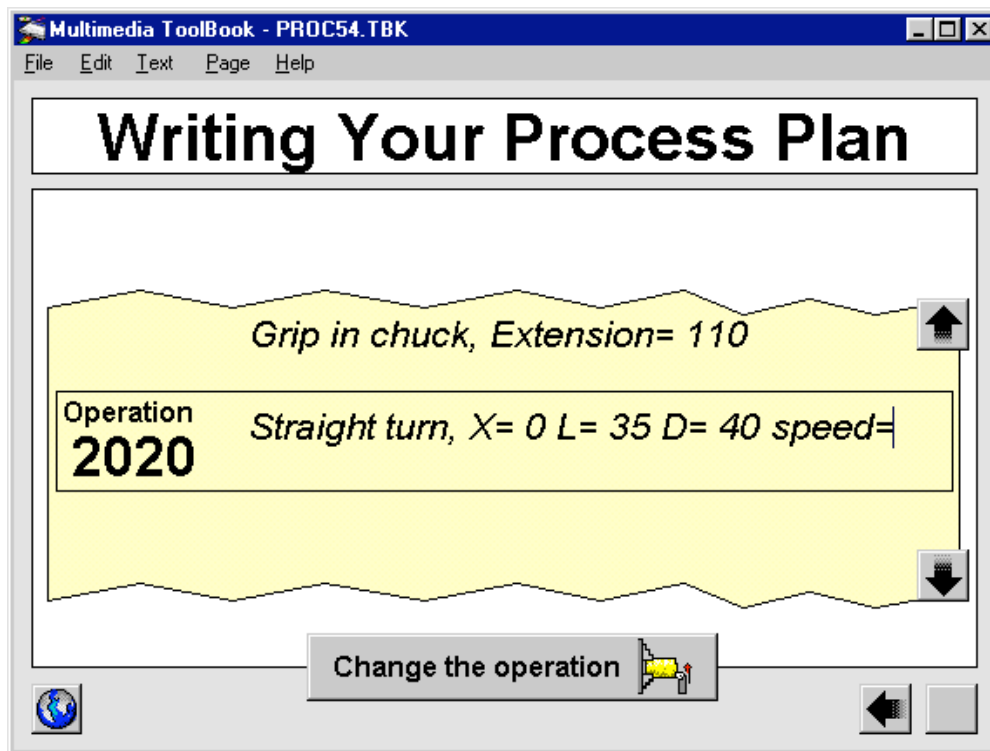


Figure 6.17: Entering individual operations

In order to produce DCG animations, CAL3P must interpret the student's intentions. In the same way, a human machinist presented with an operations sheet would read it and attempt to carry out the instructions. Of course, the computer is inferior to most human beings when a measure of 'common sense' is required to resolve a small ambiguity. Different process planners, whilst meaning exactly the same thing might nevertheless name an operation differently, ie straight turn, bar turn, turn straight, turn down, rough turn, etc. It would be difficult to create software which correctly accommodated the full variety of alternative names, and such an approach would still leave potential problems with typographical errors or misspellings in the user's input. Errors of syntax are also possible, failing to specify all the parameters which must be known for an operation to be carried out.

The solution to these problems requires the user to choose operations via a graphical menu. Instead of typing instructions directly into the process plan, clicking on the button 'Change the operation' (see Figure 6.17) takes the user to a menu which offers a number of common turning processes (Figure 6.18). In picking from this limited set, the user is forced to use terminology which CAL3P has been programmed to interpret. In fact the software also accepts a few common alternatives, but there is no guarantee that a slang term will be understood, so choosing instructions from the menu is a preferred method. Each button in

the menu includes a small graphic illustrating the named process, which should help novices.

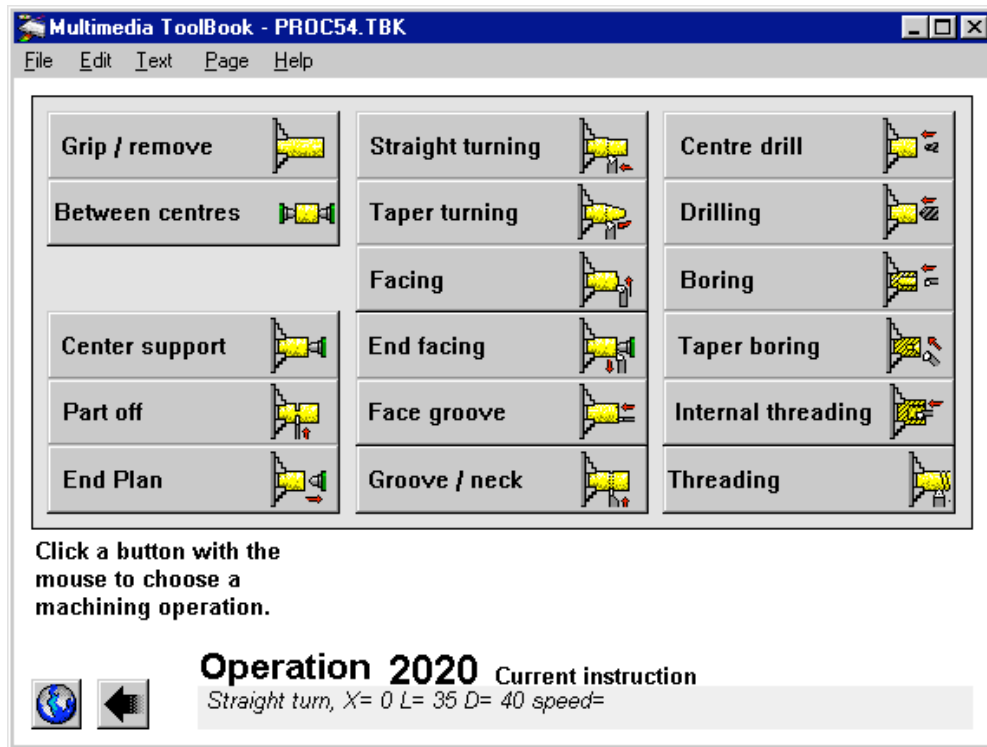


Figure 6.18: Choosing turning processes from a graphical menu

Clicking any of these buttons takes the student to a screen relating to the named operation. The user can read a short description of what the process is used for, and can specify the exact geometry which is required. Figure 6.19 shows an example – in all cases, values are entered by clicking on each ‘question mark’ button, which brings up a dialogue box asking what the new dimension or angle should be. When the desired value is entered, it replaces the question mark in the schematic.

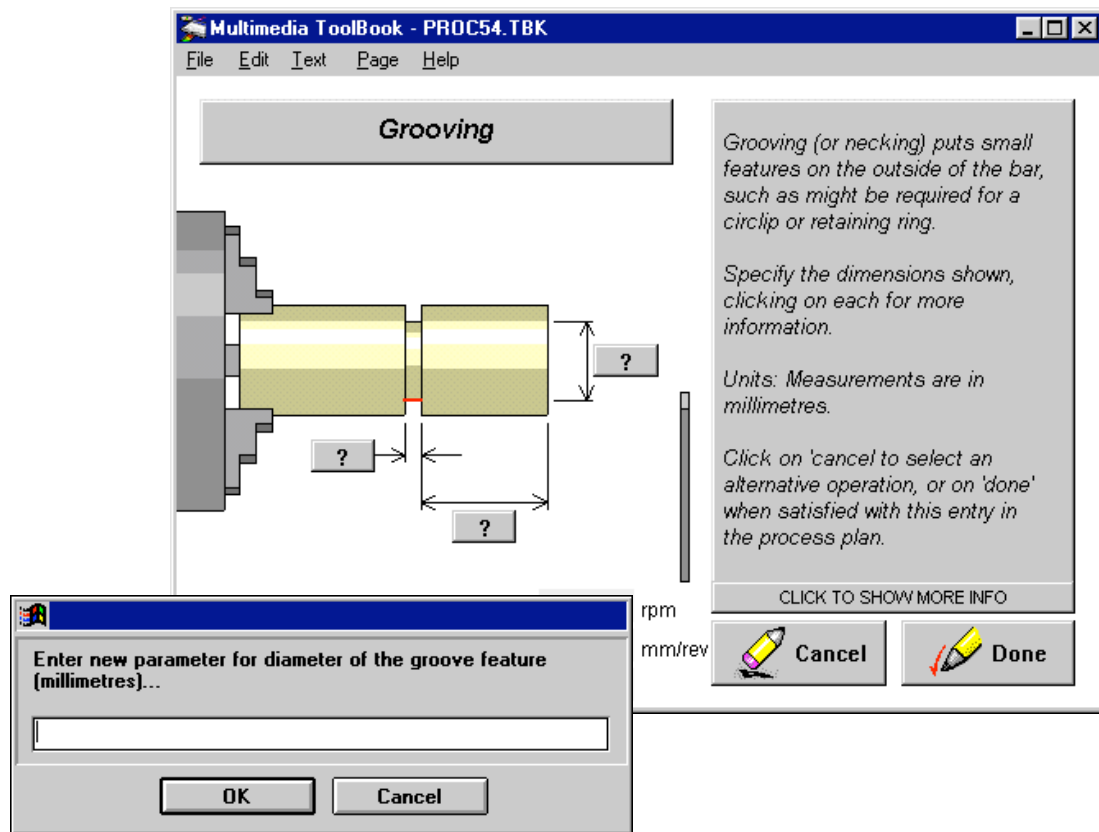


Figure 6.19: Schematic for a turning process, and an example of the pop-up dialogue boxes with which parameters are entered

6.4.3 Built-in Documentation

Where the prototype had on-line documentation which related only to the manufacturing operations which were possible, plus a small section on operating the software itself, CAL3P under Multimedia ToolBook had two new informative sections. From the main menu, the user was offered the sections 'How to use CAL3P' and 'Guide to Process Planning'. The former began by explaining the common buttons which would be used to move about within the software or access additional information. This also introduced the parameters X, L and D which users would employ to make process plans of their own (see Section 6.2.1). A final screen presented a problem from which a student could not progress until they understood how to specify operations in that format. The 'Guide to Process Planning' contained ten pages of background information about process planning, employing a mixture of text and images. When CAL3P is used following a lecture on the subject of process planning this section should not be necessary but it represents a 'safety net' for forgetful students, and allows the software to be used on its own if necessary, such as in a distance learning context.

Like the HyperCard prototype, documentation is also available on each possible manufacturing operation. These screens are seen whenever an operation is nominated, allowing the system to present detailed information just when it is required. Figure 6.20 shows one such screen. It allows the parameters required on the operations sheet to be entered (covered in Section 6.4.2), but also offers the novice an explanation of the purpose of the process.

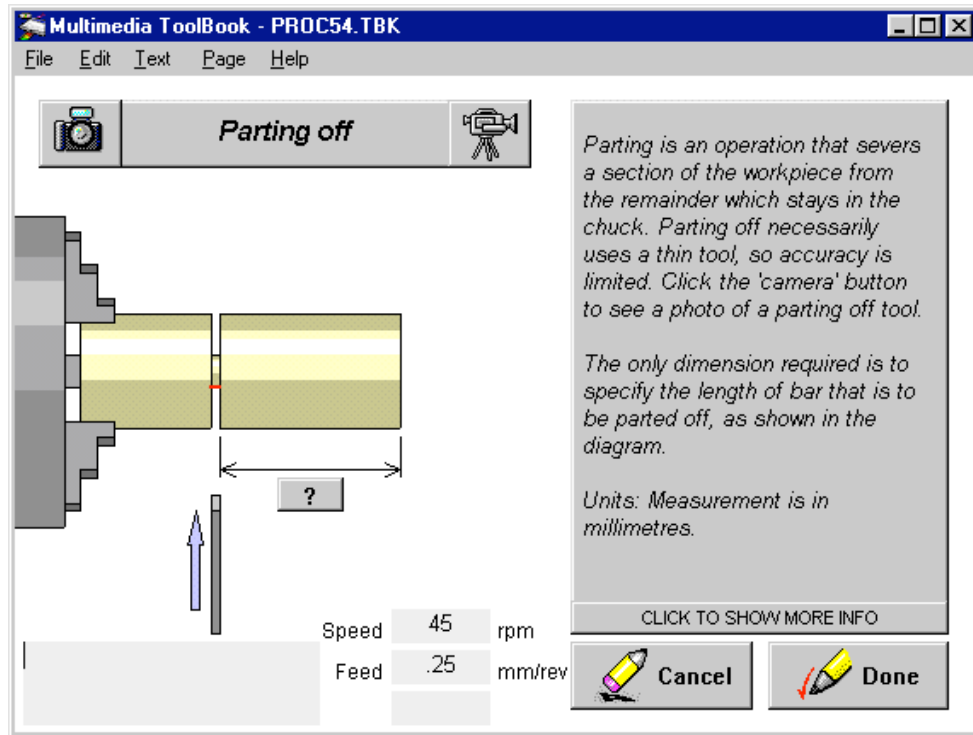


Figure 6.20: Screen for specifying a parting off operation

6.4.4 Digital Video in CAL3P

Although playing digital movies was possible within HyperCard, development of the ToolBook version of CAL3P was well underway before any movies were captured, so only that version acquired movie clips. At various places within the CAL3P software, the user would see an icon representing a movie camera. Clicking on this would bring up a short film showing a machining process.

The video clips which were built into CAL3P used only a small portion of the screen, and were not very long. They were not intended to substitute entirely for real demonstrations, nor for the use of material on video tape. However, writing a process plan in CAL3P requires that the student has some knowledge of engineering fundamentals. If this is not the case when the session begins, media such as these film clips might provide a 'safety net'. It was thought that the occasional use of video would

help a novice to relate the theory taught by CAL3P's DCG displays to real workshop practice.

The storage space required for a video file depends upon its length, display resolution, colour depth, frame rate, compression protocol and even content. Thus, it was difficult to predict exactly how large each video file would be. Table 6.1 shows the details of the six AVI (Audio Video Interleave) format files which were added to CAL3P.

Movie Subject	Duration (sec)	File size (KB)
Bar turning	26.5	3,942
Internal Boring	12.7	1,363
Centre drilling	24.4	2,830
Drilling	21.9	2,409
Facing off	17.1	2,252
Parting off	10.5	1,117

Table 6.1: AVI files associated with CAL3P

The video clips were shot experimentally under a variety of lighting conditions. The processes filmed also varied, with cutting conditions being changed in an effort to produce attractive film sequences showing detail such as coolant flow and chip formation. From all these shots, a few were selected for video capture. It was decided that six movies would be adequate for experimental purposes, though a commercial IEM product might well feature more video.

In their final form as used by CAL3P, all AVI files had the same resolution of 160 by 120 pixels, all were shown at fifteen frames per second, and a common compression method was used. The average storage space required for digital video of this specification was found to be 123KB per second.

The six video clips which were to be used were selectively shortened to show just the cutting operations themselves. Video showing the chuck starting to spin or a tool advancing but not yet in contact with the workpiece was largely eliminated in an effort to keep the video files as small as possible. Even so, the total storage space required for CAL3P's six AVI files was 13.9MB for less than two minutes of video – the rest of the CAL3P software under Multimedia ToolBook required a total of only 4.7MB. Clearly, any use of video was going to increase the storage space required for CAL3P substantially. Where it had been possible to disseminate demonstration copies of the software on a self-extracting set of four floppy disks, this approach would no longer be practical if the video clips were to be included. CD-Rom remained a possible format, though that would

confine the software to machines equipped with a CD-Rom drive, which were found to be quite scarce in computer suites in universities and schools.

It was found that the frame rates which had been used left the movies somewhat unsatisfactory; in the same way that the spokes of a rotating wheel may sometimes appear to spin backwards when seen on television, problems were experienced when trying to show digital film clips where the jaws of the chuck were seen in rotation. In some clips they appear ghostly and slow-moving, or seem to spin backwards. A similar problem could be seen when the video clip showed hexagonal bar being machined. Unfortunately, simply increasing the AVI file's frame rate may not solve the problem – unless the computer used for playback has dedicated video hardware, it may struggle to get through the file, skipping frames to keep up, and therefore looking no better. Digitising extra frames would also require a proportional increase in the amount of storage space required per second.

An alternative was investigated, employing a movie clip which was not captured from video tape, but produced within a 3D modelling package (Macromedia Extreme 3D was used). Figure 6.21 shows a still image created in this manner. It was found that video produced directly from a 3D model did not suffer from the 'flicker' problem, described above, which had been seen with real digitised video. One reason why this problem does not manifest itself may be that the desired frame rate is specified exactly when the movie is rendered – there is no interim capture stage, unlike when a camcorder was used to film the real machine tool.

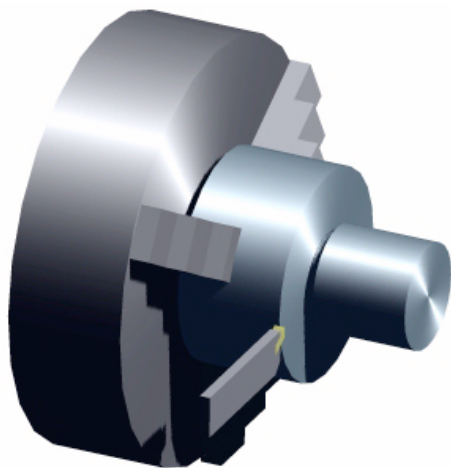


Figure 6.21: An image from a 3D modelled movie of a machining process

Unfortunately, the rendered model technique for generating movies was also found to have disadvantages; the medium looks false, with equipment appearing too clean and

bright, and the models were necessarily simplistic. A user who has only ever seen rendered movies of a generic lathe would find a real lathe confusing since it would have many more features. Even with a simplified model of a lathe, creating movie clips this way required a significantly greater investment of time than when simply pointing a camera at the real thing – development effort remains a consideration.

The experimental movies created by rendering did not convey adequately the violence of some turning operations; the coolant being splashed around, the ragged surface finish created when cutting conditions are less than ideal, or the ‘knock knock knock’ sound as the tool repeatedly strikes the edges of a hexagonal bar (there was no soundtrack produced with this method, of course). CAL3P includes just one movie which was created by this technique, showing a three jawed chuck in rotation, holding a piece of bar. This appears on the front page when CAL3P is first launched. It has the same resolution and frame rate as the AVI files, but a direct comparison with the storage space requirements of the AVI files is not meaningful; the format employs a different video compression method, the movie had different content and since it was an animation, there was no audio track. Still, at 300K for 3.2 seconds, it is little better.

6.4.5 Polygons and DCG Animation

Although, as stated in Chapter 5, almost every command in the HyperCard scripting language has an equivalent in ToolBook, the two packages handled graphics in a very different manner. In HyperCard, the generation of images is by painting. This is the creation of a picture by colouring an array of pixels. Thus, painting involves bitmapped graphics. Multimedia ToolBook can display bitmapped graphics painted or captured on other software, but its internal graphics capability is centred upon drawing – the creation of a picture in vector graphics, typically by building up layers of polygons. This technique had already been employed when creating slideshow for the DICTA project under Microsoft PowerPoint (see Chapter 5). Multimedia ToolBook’s own use of bitmapped graphics was confined to the creation of icons for buttons.

Some of the commands employed in the scripting of animated vector graphics were given in the previous chapter (see Section 5.7.1). The technique was similar here, though where that animation is ‘hardwired’ to look the same every time it is played, variables were used in the CAL3P scripts to generate content which reflected the student’s process plan. An example is shown below:

```
if word 1 of currentop = "grip" or word 1 of currentop = "clamp"
                                                                    then
    extn = word 5 of currentop
```

```

draw rectangle from (xat+(extn*vsc)),(yat-(matrad*vsc)) to
                    (xat+((extn-matlen)*vsc)),(yat+(matrad*vsc))
name of selection = "bar"
rgbfill of selection = 191,191,128
i = 1
loctx = item 1 of the position of irregularpolygon "topjaw"
locty = item 2 of the position of irregularpolygon "topjaw"
locbx = item 1 of the position of irregularpolygon "botjaw"
locby = item 2 of the position of irregularpolygon "botjaw"
pause 25
while i <= 15
    position of irregularpolygon "topjaw" = loctx,(locty+10*i)
    position of irregularpolygon "botjaw" = locbx,(locby-10*i)
    i = i+1
    pause 5
end while
end if

```

The (simplified) script segment shows how polygons are drawn by specifying their coordinates. The 'while ... end while' loop makes the irregular polygons representing the jaws of the chuck appear to grip the bar, but this is not just a generic example of the process; the bar which is shown is of the diameter specified by the student, and it is gripped with extension as specified. Subsequent animated machining operations likewise reflect the process plan.

6.4.6 Scale and Animations

Under Multimedia ToolBook, CAL3P supported several new turning operations such as taper turning, increasing the capabilities of the machine simulated so it could be used for planning a wide variety of turned parts. One important question, raised during an industrial demonstration, concerned the working 'envelope' of the machine tool it simulated. What size of part could be manufactured? In the HyperCard prototype, the screen itself imposed a limitation on the size of part which could be represented. A capability to scale the screen display was added. By adding a multiplier to every coordinate used in the DCG animations, it is possible to scale the display up or down as required. In the script example given in Section 6.4.5, notice the 'vsc' variable; this was programmed into every calculation relating to a polygon's position or that of its vertices. Theoretically, with this feature CAL3P allows the representation of any part from a few millimetres in length and diameter such as a delicate piece of a watch mechanism, to a massive part such as a ship's propeller shaft. For a large part, it is possible to use the scale function to zoom in and examine small features which would not be visible on a view of

the whole part. This is not realistic because no single machine tool in the real world could accommodate such a variety of product sizes, but it enables CAL3P to be useful in as wide a variety of situations as possible. Of course, the advantage with ToolBook's vector graphics is that a feature does not look increasingly rough when one zooms in, as it would if CAL3P used bitmapped graphics. In the manufacturing simulation screen, clicking the button 'Prefs' brings up scale controls, allowing the user to select an appropriate view.

6.4.7 The Built-in Quiz

Later versions of CAL3P under Multimedia ToolBook included a built-in quiz of ten questions which a student should attempt to answer after working through an introductory process planning exercise. This feature serves as a reinforcer for the learner, but most importantly it was designed to gather experimental data on the amount of learning which had taken place while the software was employed. The quiz and its results are detailed in Chapter 7.

6.5 Expanding the Scope of CAL3P

The software detailed in Sections 6.3 and 6.4 addresses the generation of an operations sheet for a turning process. In truth, the process planner may begin to make important decisions at a higher level. At this macro process planning level, Jasperse [1994] identifies tasks including process selection (how to achieve a given feature), then workstation selection (which machine or cell to use), and then setup planning (choosing the basic sequence in which the part will be made, i.e. ordering the entries on the route sheet). Thus, before the specification of machining operations begins, the process planner has already made a number of decisions concerning the capabilities of the factory. A prototype module of CAL3P explored the possibility of interactively teaching the student about this strategic level of process planning. Products are typically made up out of sub-assemblies and/or components. These must be bought or made, and those which are to be made require detailed process planning. A method of manufacture will be determined, to work with a casting or machine from solid material for instance. These decisions will ultimately be reflected on the route sheet. Only at the next stage were students required to become involved with process planning under CAL3P, generating detailed work instructions in the form of an operations sheet.

In order to give the learner a better appreciation of the engineering issues, of which generating an operations sheet in CAL3P is just a part, the prototype software module

presented a case study to the student. This involved the manufacture of a gear pump; an assembly of three castings, two largely rotational parts and some bought-in fasteners.

The case study initially introduces the whole product (Figure 6.22), and then breaks this down into its component parts, eventually demanding the detailed process planning of only a single part. That part is the primary shaft, which can be made by turning a blank, cutting the gear teeth and milling a keyway. Alternatively, the primary shaft could itself be a subassembly (Figure 6.23).

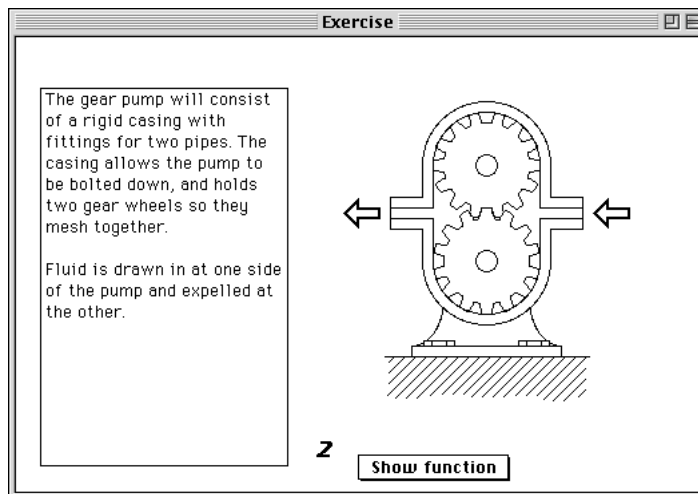


Figure 6.22: Introductory screen display from the case study

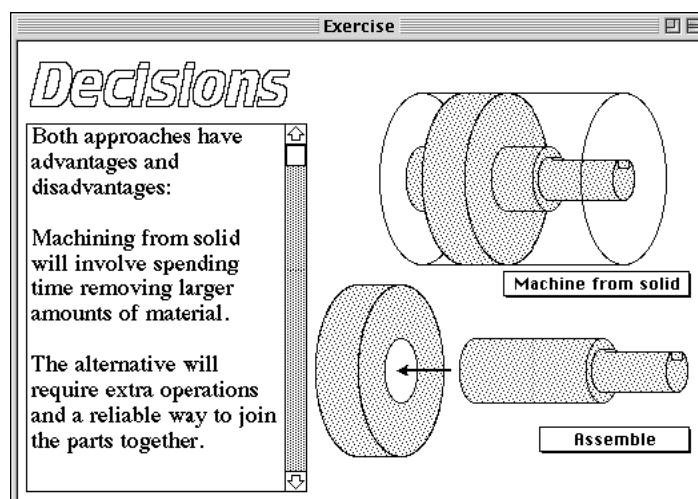


Figure 6.23: Example of the decisions to be made by the student in the case study

By reading through the case study, the student finds out the exact nature of the product, and begins to make the decisions which will influence its manufacture. He is able to ask a number of preprogrammed questions, finding additional information that he

thinks might be relevant. Figure 6.24 shows a screen where buttons could be clicked to 'ask' for information. Parkinson et al [1996, 1997] similarly allowed the user to ask questions before commencing a design exercise, but required the user to enter their questions in plain English. This excellent approach means that the student really is required to get involved with the exercise if they are to find out the things they will need to know because they are not led into asking questions simply by clicking on buttons. Parkinson's 'IDER' software [1996] interprets users' questions by pattern matching, looking for commonly asked words and phrases and responding with standard answers. Of course, there is a finite set of preprogrammed responses which the software can make, and therefore the chance a student might ask a question the system cannot answer.

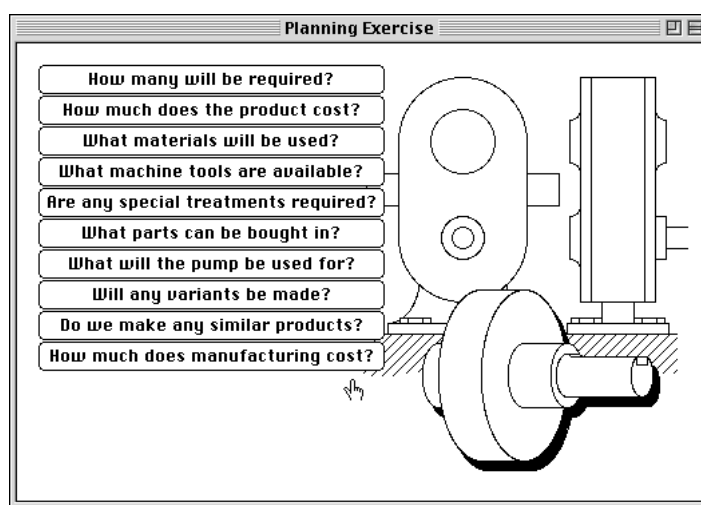


Figure 6.24: Screen from the process planning case study allowing additional information to be gained

Ultimately, as the student reads through the case study material, he will have learnt enough about the part to propose how it might be made. With all the decisions made at the strategic level, the user has learnt enough information to fill out a route sheet for each part. A graphical model of this information was made, again a DCG representation. This showed a 'virtual factory' where a number of work centres are shown, representing the capabilities of a small manufacturing business. Lines show the route a part would take through the factory (Figure 6.25). In addition to drawing lines between the workcentres which will be involved in the manufacture of each part, a simple animation shows an icon of the part moving along the route defined by the user. This removes any ambiguities which the diagram itself might show if the route sheet required that a part visited a workcentre more than once.

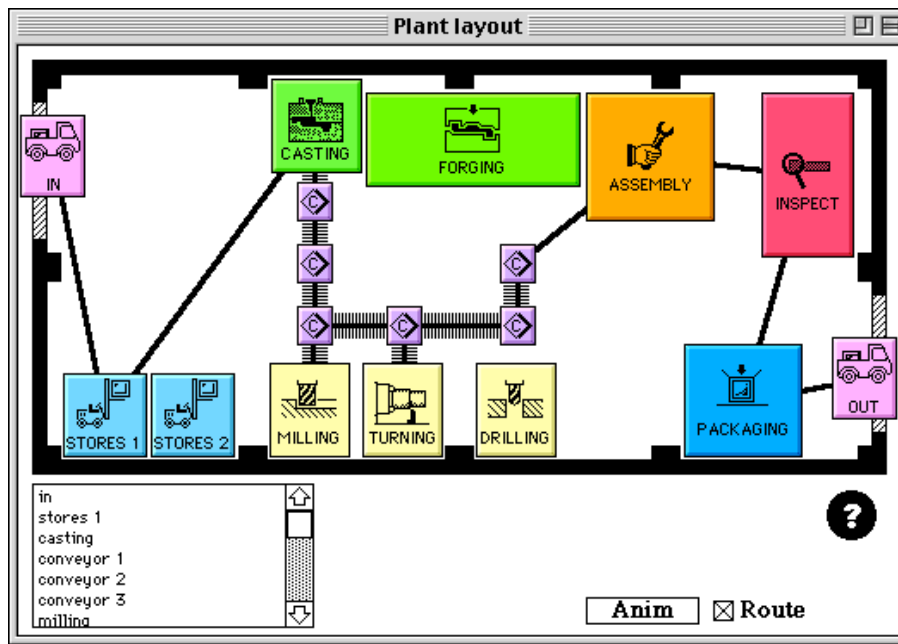


Figure 6.25: Graphical representation of a product's route through the factory, as defined in the student's route sheet

Each of the icons representing a workcentre was also a button, it being intended that the factory layout acted as a menu where the student could select to zoom in on individual processes for which detailed process planning would be carried out. For example, the 'turning' button linked with the HyperCard prototype of CAL3P, allowing the user to create an operations sheet to go with the routing already specified.

If this macro process planning approach were adopted, it would require a set of modules like CAL3P, each allowing process planning for a different machine tool. For teaching the fundamentals of process planning, it had been a deliberate choice to start with turning because it was by far the easiest common machining process to represent with DCG animations. The machining of prismatic features, produced by tools such as milling and shaping machines, could not be adequately represented with two dimensional graphics of the kind CAL3P employs.

It had been seen CAL3P's DCG approach involved greater development effort than is normally associated with multimedia software. Expanding CAL3P to cover all aspects of macro process planning in a similar way would require a further increase. In its current form, CAL3P supports a wide range of turning operations even though it is unlikely that a student would use all of them. Similarly, a 'Macro CAL3P' package would have to allow process plans to refer to a wide range of machine tools, even though only a few of these will be used in the manufacture of any one part.

The macro process planning case study had taken the alternative approach of specialisation. The case study was programmed to guide the user through a series of decisions, arriving at a strategic plan for the manufacture of a single part. It was thus a 'hard-wired' exercise – the case study could not be used in a variety of different ways. This went against the principle of CAL3P; DCG was envisioned to allow software to be used repeatedly with no loss of educational effectiveness, but the case study module could only really be worked through once or twice.

The process planning case study was demonstrated to engineers at Rolls Royce Aero Engines in Derby. The turning simulation was well received, but the case study as a whole was considered unrealistic because it gave the student more freedom than a real process planner would have. Where the student could choose to cast, forge, fabricate or machine from solid, a real process planner would receive a part drawing with such details already established. As a result, the case study software has never been used with students.

6.6 Limitations of CAL3P

Development effort requirements, audience expectations, capabilities of the delivery hardware and the syllabus of the learners all impose limitations on IEM software, and CAL3P is no exception. It is hoped that the use of DCG has shown how a system can meet a variety of different user requirements while making only moderate demands on computer hardware.

There remain processes which CAL3P does not support. It is impossible to specify curved profiles and see them made within the software. The line in an operations sheet for a complex profile would have to say simply, 'Turn to profile A in drawing' – it is not possible for the current version of the software to respond to such instructions. CAL3P exercises do not require such features.

All CAL3P's turning processes involve generating, not forming (the use of specially shaped tooling to create complex profiles with a single pass of the tool). There is no support for the creation and use of form tools, but the lack of this feature was not considered detrimental to the software since it forces the student to work through the exercise, detailing a series of individual cuts to generate the profiles required.

Speeds, feed rates and cutting depths have not formed a part of any exercises to date. Space was made for these parameters to be entered in a process plan, but in the prototype these fields were hidden by a panel marked 'advanced features disabled', to simplify the user interface as far as possible (see Figure 6.13). It was considered that specifying the machining operations themselves would constitute enough of an exercise;

cutting conditions are a related topic for novice engineers, but not one which is necessarily taught at the right time to feature in the process planning exercise.

If cutting conditions were specified by the student, it would be necessary for CAL3P to check them for realism. This is a very complex issue in itself; Petty [1983] created software which would optimise cutting conditions to achieve given geometry, taking into consideration such issues as the propensity of the workpiece to deflect, the power rating of the specified lathe and the useful life of a tool, as defined by Taylor's tool life equation [1907]. Thus, to assess the viability of a specified machining operation, calculations of this kind would have to be built into CAL3P to compare the students' calculations against the optimum. Adding this feature to the software would be a project in itself. Instead, a simplification was made, based upon the volume of material removed per second. If the depth of cut (d) and the feed rate (f) are known, the cross sectional area of tool and workpiece which are coming into contact during machining can be calculated (figure 6.26).

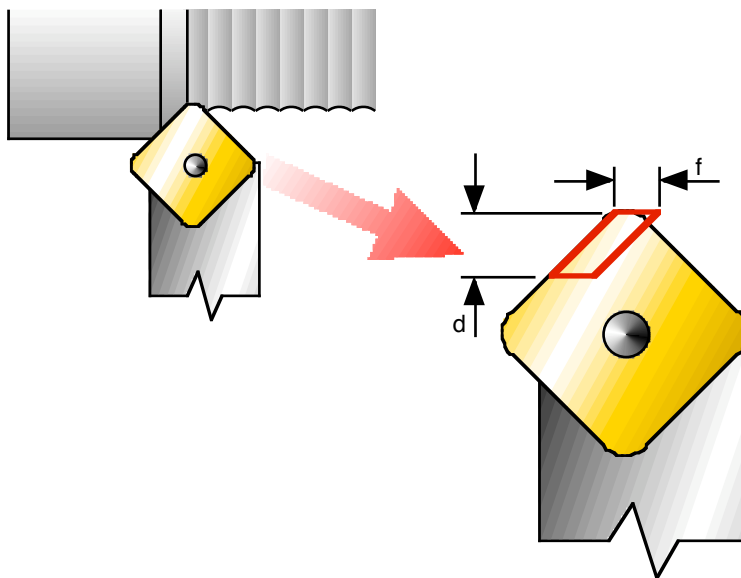


Figure 6.26: Simplified interpretation of turning parameters

If the speed of a machining process is specified as a constant surface speed in metres per minute, as is common for CNC lathes, a conversion of this to millimetres per second (c) allows the total volume of material removed per second to be calculated as $d \times f \times c$. For each of the materials that a student can specify for use within the process plan, a different value for 'allowable maximum material removed per second' exists. When the student's cutting conditions cause $(d \times f \times c)$ to exceed this value, a failure occurs. In reality, the tip of the tool would snap off or melt.

This is a tremendous simplification, but it allows CAL3P to check entries quickly, ensuring that they are broadly realistic. It does not really take into account tool life, only breakage, and does not test for factors such as coolants being used, it being assumed that a machinist would use coolants and lubricants when appropriate.

Within CAL3P the checking of cutting parameters was only implemented for straight turning operations, due to time constraints. This feature functioned adequately, and could be applied to the other turning operations supported by the simulation.

6.7 Conclusions

CAL3P software provides a useful tutorial on the elements of process planning and reinforces theoretical knowledge with a practical application. The flexible, interactive nature of the software was seen to be useful to students with a variety of different learning styles, for instance being able to read theory in detail first, or learn experimentally.

Trials with CAL3P have made a powerful case for the use of DCG animated graphics, as distinct from digitised video or similar inflexible resources, reinforcing the learning process by demonstrating the result of the machining operations as specified.

In teaching process planning to novice engineers with CAL3P, the role of the tutor became more interesting in that the software checked for clerical errors in the plan, leaving the supervisor free to address general topics of engineering practise and to elaborate on issues raised in the tutorial. It was still useful to have the tutor present, to comment on engineering problems which might arise – CAL3P's checks are not exhaustive.

It was seen that the ability to explore various possibilities with the system helped to maintain students' interest. This 'ownership' of the content being displayed appears to have been significant in bridging the gap between an abstract theoretical exercise and practical experience [Farr and Lawlor-Wright, 1996].

Learning theory supports the hypothesis that improved learning of process planning took place when CAL3P was employed; simulation offers a rapid form of feedback for the learner, and therefore more effective reinforcement. Graphical representations are more easily understood and encoded. Describing an operation such as 'facing off' in text is much harder than showing the process being employed. Since machining involves moving tools and workpiece geometry which changes as operations are carried out, animations or movies ought to be superior to images in a textbook. As seen in Chapter 3, Buzan [1993] identified that vivid media were more easily remembered, and that other peaks in attention occurred when the information being taught was related to something

already known. Since the animations generated are based on the plan 'owned' by the student, it qualifies here, ensuring that the student watches the machining simulations and learns from them.

Though the initial trial of CAL3P had resulted in a favourable reaction from students, it was still desired to gain a clearer indication of its value. More detailed testing was planned which, it was hoped, would show the presentational approaches which were most effective within IEM software. One of the approaches to be evaluated was the DCG technique itself. In Chapter 7, the tests which were carried out are reviewed.

It has been seen that CAL3P has required, in places, considerably more development effort than conventional IEM software. Chapter 7 will endeavour to show if the effort inherent in the provision of the DCG elements has 'paid off' in terms of increased educational performance.

It must be remembered that CAL3P is a 'test rig' rather than a finished product meant to compete with existing, commercially produced IEM software. Thus, it addresses a limited section of the work of the process planner, and is in some places incomplete or oversimplified, as described in Section 6.6. The total development effort for the whole package is a misleading figure, but the development process for the provision of individual sections was monitored, and is discussed in Chapter 8, contrasted with the results of tests to establish the effectiveness of teaching when using the various approaches demonstrated within CAL3P, such as diagrams, still photos, animations, movies, text and DCG animations.

7.0 Experiments Conducted

Chapter six described the development of CAL3P, a piece of multimedia software which was to be used not only in a teaching function, but also to serve as a testbed for experiments. This chapter describes how a built-in quiz was used to measure how much the students had learned from the software.

CAL3P was used in experiments aimed at determining the value (in terms of improvement in student skill level arising from use of the software) of several possible approaches to the provision of computer-based information. Chapter 4 described how various levels of interactivity are possible within IEM software; the experiments which are detailed in this chapter were conducted to gather data on the educational effectiveness of various media and levels of interactivity. With these data, a comparison between the effectiveness of a software feature and the effort inherent in its provision became possible. Thus, the creation of CAL3P can itself be considered to have been an experiment aimed at supplying data such as the time spent in the development of each aspect of the software.

Chapter 1 introduced the issue of software development effort, suggesting it might be a crucial factor in the viability of a multimedia software project. The Inter-University Committee on Computing [1991] reported that in a survey on CAL, 85% of universities claimed they lacked the resources necessary to develop applications. Despite this, CAL was seen as one way of coping with an anticipated 50% increase in student numbers (if funding was available).

Ultimately, bearing in mind that the hundreds of hours' work involved would resource a substantial amount of conventional instruction, it may be that the development of a piece of IEM software can only be justified if it can be proved that an IEM approach will:

- be superior to conventional methods in terms of the results obtained,
- be cheaper overall than conventional instruction,
- allow learning to take place where it would otherwise be impossible.

It was hoped to find out if CAL3P – a piece of IEM software with DCG elements – qualified as better, cheaper and/or more accessible than normal classroom contact.

7.1 Assessing the Effectiveness of CAL3P Instruction

During the development and testing of the CAL3P prototype (detailed in Sections 6.3 to 6.3.2), observation and questionnaires had suggested that the software offered students an enjoyable exercise, but it remained necessary to show that the exercise had caused learning to occur. As section 3.5 described, the amount of learning which takes place cannot be measured directly. It must be inferred from the skills and knowledge demonstrated subsequently by the learners. This requires that tests are set which show when transfer and recall take place. Such tests were first applied when a group of undergraduate students worked through a CAL3P exercise in April 1997.

Initially, experimental student attainment testing was carried out by following the CAL3P tutorial with a short written test. A copy of the test paper is included in Appendix 4. This method was found to be unsatisfactory; students' first reaction was to get the answers they needed from the computer rather than from memory. The students were still in the computer lab, and surrounded by others who were still using CAL3P, so referring to the software was an easy way to find answers. Though this tactic suggests that the students had noticed enough while using the software to know that the answers could be found within, it was otherwise disappointing. This sporadic 'cheating' by looking answers up on the computers rendered the experimental results dubious, so they have not been used in any evaluation of the software. It was found that the written test also generated a significant amount of work, in marking the students' papers and returning them with comments. In addition to placing an additional load on the tutor, marking caused a delay in the feedback loop, reducing the educational value of the test for the students.

Following this experience, an alternative approach was tried, with CAL3P modified to include a built-in test module. Students were required to work through a process planning exercise as normal, but they were then directed to enter the 'CAL3P Quiz'. Clicking on the appropriate button in the main menu brought up a warning that once the test was started it would be impossible to return to the informative sections of the software. This limited each student to recalling what they had learned, not hunting through the software to find the answers they needed. Students were told to start the quiz once they had completed the process planning exercise. Once each student had done the quiz and read the model answers to any questions they had answered incorrectly, they were dismissed.

Trials of CAL3P which made use of the built-in quiz took place from December 1997 onwards, as shown in the chronology presented below:

March 1995	Process planning was taught to undergraduates by a conventional lecture and then a paper-based group exercise during a 3-hour tutorial. Marking the groups' work revealed that some students had little understanding of machining processes.
December 1995	M.Sc. group used the HyperCard prototype in a 1-hour exercise, following a conventional lecture the day before. Their reactions to the software were largely favourable (detailed in Section 6.3.2).
December 1996	An M.Sc. group used an early ToolBook version of CAL3P, with interactive DCG simulation, but no movies or photos. The primary purpose at this stage was to test the new user interface.
April 1997	Undergraduates were given a 30 minute lecture on process planning, followed by a 90 minute session on CAL3P and a short written test. Their evaluation of the software was favourable, but the test results were unsatisfactory (as described earlier in this section).
December 1997	Eighteen M.Sc. students were given a short introduction to process planning in a conventional lecture, followed by an hour on the software, and ten minutes on the new built-in quiz. Twelve sets of quiz results were obtained; scheduling limitations in the computer lab prevented the last group of six students completing the quiz.
March 1998	Undergraduates (B.Eng. Manufacturing Management and B.Eng. Manufacturing Engineering) followed a lecture with up to two hours on CAL3P, all completing the quiz and an evaluation form.
April 1998	Undergraduates on a part-time B.Sc. Manufacturing Systems Design course at Bolton Institute of Higher Education worked through the standard CAL3P exercise, and completed an evaluation form designed by their course tutor.

7.1.1 Design of CAL3P's Built-in Quiz

Quizzes are a common feature of many CAL packages, typically intended to test how much the student can remember at the end of an informative section. This gives the student an idea of how much they have learned, and may be used to prevent progress to new sections until a certain standard is reached. Likewise, CAL3P's quiz provided an informal way for students to identify any weaknesses in their understanding of the material they had just worked through. The quiz thus served a dual purpose; while helping the student it was possible to gather experimental data in an unobtrusive manner.

CAL3P's quiz consists of ten pages of questions, broadly based on the earlier written test, each making use of text and diagrams or photos, with the exception of Question 7 which features a video clip. Since some of the questions are in multiple parts, a total of nineteen separate elements are tested over the ten pages.

The questions in CAL3P's quiz require three different kinds of interaction from the student; missing word, multiple choice and drag-and-drop questions. Missing word questions give a factual sentence with a key word left out, requiring that the student types a response into a text box indicated by a flashing cursor. Multiple choice questions take the form of a pop-up list from which one line can be selected. Drag-and-drop questions are also effectively multiple choice, but involve dragging a graphical element into place on a diagram.

Each question was designed to test for information which could be acquired from a distinct source. For example, to answer one question might have required the reading of a page of text, while the answer to another might be learnt from a simple diagram. Thus, by measuring quiz performance over a number of questions, it was hoped it would be possible to quantify the contribution that each medium made towards learning in this field.

Students' answers were recorded within the software, and could be inspected subsequently by the author. It was not simply the student's total score over nineteen questions which was of interest, but the individual answers which had been given. In addition to scoring a student's answer as correct or incorrect, by storing the actual answer given, CAL3P allowed the amount of learning which had taken place to be better inferred. This required a subtle combination of questions. It is difficult to prove the source of a single correct answer; a student may have known the answer before being taught, or they could have made a lucky guess. Hence, tests included several related questions meant to expose where answers were guesses. Where a student consistently answered related questions correctly, it could be assumed that they understood the procedure being tested.

It remained difficult to prove whether a correct answer had occurred as a result of the teaching which was designed to impart the knowledge, or if the material was already understood. This was the most significant covariant in the quiz experiment, and a particular problem for the validation of CAL3P in general. It was necessary to demonstrate that learning had taken place under CAL3P, not simply that students could demonstrate a certain level of attainment. From March 1995, when process planning was taught by purely conventional means, observation of the students has indicated a tremendous variety in their level of knowledge in engineering fundamentals. Some students had only undergone an introductory module in their first year while a few had

previously worked in manufacturing and were familiar with machine tools and work instructions.

Ethically, it is very difficult to divide a body of students and give them instruction by different methods. It was not possible to teach some of the students conventionally and others with IEM software, so a comparison of CAL3P against a conventionally taught control group was not an option. The students' timetable also prevented this approach, and the total number of students in each class was small. Further splitting them would have made the results subject to considerable variation.

To take the main covariant into account and ameliorate its effect, students were broadly identified as belonging to one of two groups; experienced engineers and novice engineers. In order to identify the experienced engineers, special questions were included within the quiz. These were meant to test for knowledge which an experienced engineer should have, but which CAL3P did not teach. Only a person who had previously worked with machine tools or studied their operation beyond the level required in the students' courses would be able to supply correct answers. These 'trick questions' were embedded among the others, and are identified in section 7.3 as they occur.

It was important to make this distinction since it would be unreasonable to claim that a piece of IEM software had taught students a skill simply because they were able to demonstrate it. Thus, those identified as experienced engineers were not considered to have learned from CAL3P simply because they returned a high percentage of correct answers, but those whose poor performance on the 'trick questions' identified them as novice engineers were assumed to have gained knowledge through using the software, where they scored well on the built-in quiz. Figure 7.1 shows that there were two possible outcomes for each kind of student.

		Student	
		Novice	Experienced
Result	Correct	1	3
	Incorrect	2	4

Figure 7.1: Possible question outcomes

Where a novice supplied the correct answer in a quiz (1), it suggested that CAL3P had succeeded in bringing him to a position where he could supply the correct answer to

a question about process planning. Where an incorrect answer was supplied (2), CAL3P had failed to teach that fact to the student. For an experienced engineer – if the method of identification was valid – correct answers should be the norm (3). Though these correct answers could not be taken as proof that the software’s teaching was effective, their presence in quantity would indicate that an experienced engineer had been correctly identified; conversely, a high number of mistakes on the part of a person identified as an experienced engineer (4) would suggest that the method used to categorise students was incorrect. It was anticipated that only results (2) and (3) should be common for the ‘trick questions’.

7.2 Experimental Method Employed

For CAL3P, the learning process tested concerns the principles of process planning, with worked examples relating to turned parts. In all trials within the University of Salford, an introduction to process planning was given in the form of a conventional lecture. Use of the CAL3P software followed within 48 hours. Students would spend about 60 – 80 minutes using the ‘informative’ section of the software before moving on to the quiz, which required about 10 minutes to work through, but was not a timed exercise. The answers supplied by students in the quiz were used to infer the amount of learning that had taken place.

Experiments to measure learner attainment will necessarily involve a number of variables which have an impact on the performance observed. The most significant variable has already been identified in Section 7.1.1 as the existing level of competence of the learner. For this programme of experiments, some variation was deliberately introduced, such as the differing mix of media which was employed to convey various facts in the informative section of the software. This was important because it was aimed to discover which media will be of the greatest value in future computer based learning materials. The desired result – improved student retention of information – should then be achievable by designing future CAL software to feature the qualities identified.

7.2.1 Knowledge Tested

CAL3P allows process plans to be written for almost any turned part. In providing a capability to represent a wide variety of turning operations, only a subset of these would be employed during a single exercise. It was clear, therefore, that the CAL3P software would not always be used in a predictable, linear manner. Despite this, it was necessary to understand what information the students were likely to have seen – and what they

would have attempted to learn – by the time they reached the quiz stage. To some extent this can be determined from the geometry of the part for which the students were required to write a process plan.

CAL3P student exercises took the form of a part drawing for which manufacturing instructions were required. From the part drawing it is possible to deduce which machining processes the students are likely to have employed, and therefore which sections of CAL3P they have studied in detail. The amount of time spent on the CAL3P exercise must also have been a function of the complexity of the part drawing. (The exercise which was used with students from April 1997 onwards is shown in Figure 7.2).

It was hypothesised that questions relating to the processes which students had learned about interactively (by specifying them in their own process plan and observing the results) were more likely to be answered correctly in the quiz.

Process Planning Tutorial

Use the CAL3P software to write a process plan for production of the part shown below. It will be produced on a CNC lathe in occasional batches of 100. Aluminium suitable for the job is available in 105mm diameter bars which can be cut to any required length with a billet saw. A good process plan will use a minimal number of setup operations, limit tool wear as far as possible and produce parts of the right geometry at a reasonable rate.

When you have completed a process plan, test your work and make any changes necessary until you are satisfied. You should then do the short computer-based quiz which can be accessed from CAL3P's main menu.

Drilled hole $\phi 20$
x 55mm deep

Dimensions: millimetres
Material: aluminium

Figure 7.2: Student process planning exercise

A variety of different approaches can be taken to the machining of the part shown in the student exercise. While there can be no single model answer, it can be seen which operations a student who has completed the exercise ought to have experienced. A student will enter manufacturing instructions to specify that material of suitable length and diameter is issued from stores, and then grip the bar appropriately. A good process plan will begin machining with a facing operation to put a neat end on the bar. Externally, the part consists mainly of simple 'step' features which can be made by straight turning, but a straight turning approach will be inappropriate for the groove feature 'A' (see Figure 7.3). The conical feature 'B' will require a taper turning operation. At this point the workpiece may be cut off from a larger section of bar with a parting off operation. At some point it will be gripped in the chuck the other way around to allow the rest of the machining operations to be carried out, such as the stepped feature 'C'. The central hole will be made by drilling, for which centre drilling must first be carried out. (Appendix 5 shows one possible set of manufacturing instructions for the part.) From the exercise undertaken, the machining operations which the student should have encountered before reaching the quiz can be predicted. Sections 7.3.1 to 7.3.10 detail each of the questions in the quiz, identifying questions pertaining to skills which the students should have acquired while doing the exercise.

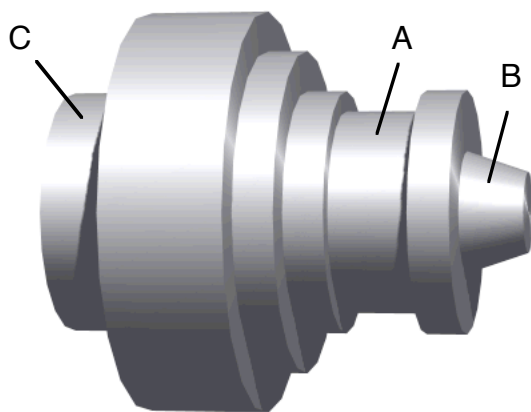


Figure 7.3: Finished geometry of the workpiece in the student exercise

7.3 Questions in the CAL3P Quiz

The scenario in which students are required to work through the quiz has been described. In the subsections that follow, the questions which each student is asked are

detailed, showing the screen display employed, listing possible answers and identifying the response(s) which CAL3P marks as correct.

Before entering the quiz the student is shown an introductory screen which explains the kinds of interaction which will be required (Figure 7.4). This familiarises the student with the interface before he is required to answer a marked question.

The quiz is entirely linear; it is not possible to go back and change an answer once the student has moved on. This is necessary because some of the questions use terminology in their examples which might otherwise supply a student with answers to earlier questions. Question 3, for example, requires the student to name some parts of a lathe including the chuck. The chuck itself is the subject of Question 8.

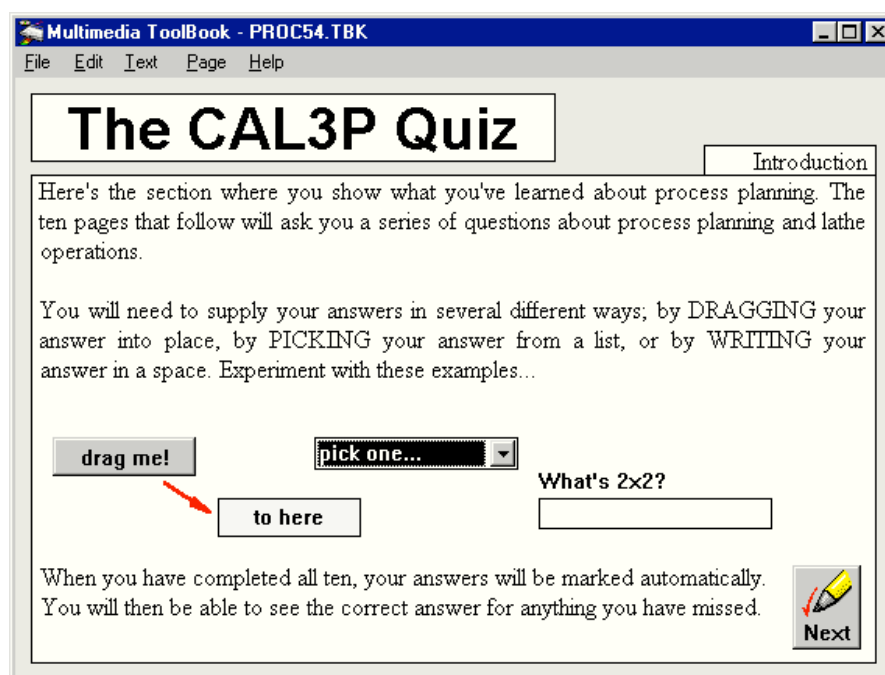


Figure 7.4: Introductory screen to the quiz

The student is offered no indication as to whether answers are correct during the quiz. Regardless of the answer he has supplied, a simple 'beep' is played and the computer display advances to the next screen. Meanwhile, the student's response is stored in a hidden page, and their answer is marked as right or wrong on a summary page to be shown once the last question has been answered.

7.3.1 Question 1: Drilling

Question 1 was aimed at detecting a common misconception which some students had been seen to make. It had been observed when testing prototypes of CAL3P that some

students falsely referred to drilling operations as 'centre drilling'. Drilling on a conventional lathe is carried out by holding a drill rigidly in the tailstock, and sliding this up to meet the rotating workpiece. Naturally, holes can only be made in line with the part's axis of rotation; this may have led some students to call the operation centre drilling, a process which involves making a small indentation with a special drill, often used to act as a guide for a subsequent drilling operation.

This is a fairly fundamental question for anyone with practical machine tool experience, but some students in 1995 (before CAL3P was introduced) had included erroneously named drilling instructions in their process plans.

In carrying out the CAL3P process planning exercise, the students had been required to do some centre drilling and drilling. When these operations were chosen, the students were able to read text describing their purpose. CAL3P's DCG animations demonstrate both drilling and centre drilling.

A multiple choice question was designed which offered four possible answers; centre drilling, drilling, boring, reaming. One answer could be selected with the mouse, and dragged into place (Figure 7.5).

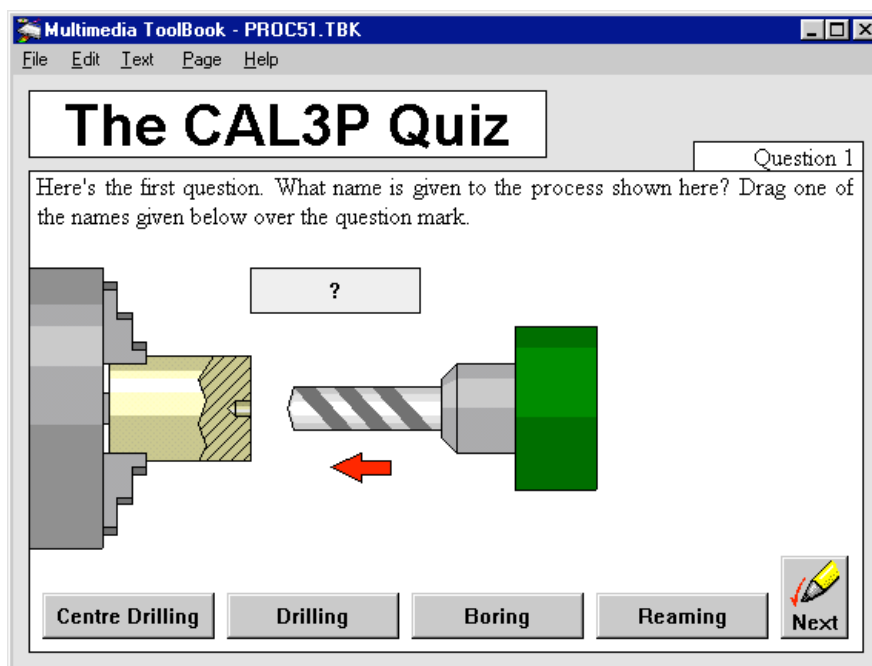


Figure 7.5: CAL3P quiz, question 1

7.3.2 Question 2: Internal Boring

Question 2 was intended to assess the student's ability to select an appropriate machining process and name it correctly. The question took the form of a sectional view of

a workpiece, such as the students should have been familiar with from their experiences in the planning exercise (figure 7.6). A part drawing is shown, where the completed product can be seen to have an internal rebate. The student is asked to name the process which should be used to complete the part. The options are given in the form of a pop-up list; slot milling, internal boring, taper turning, grooving, finishing. Internal boring is the correct answer.

Internal boring was not a required operation in the process planning exercise which the students had done, so quiz results among novice engineers were expected to be lower than for processes which had been employed to achieve the part geometry required in the CAL3P exercise.

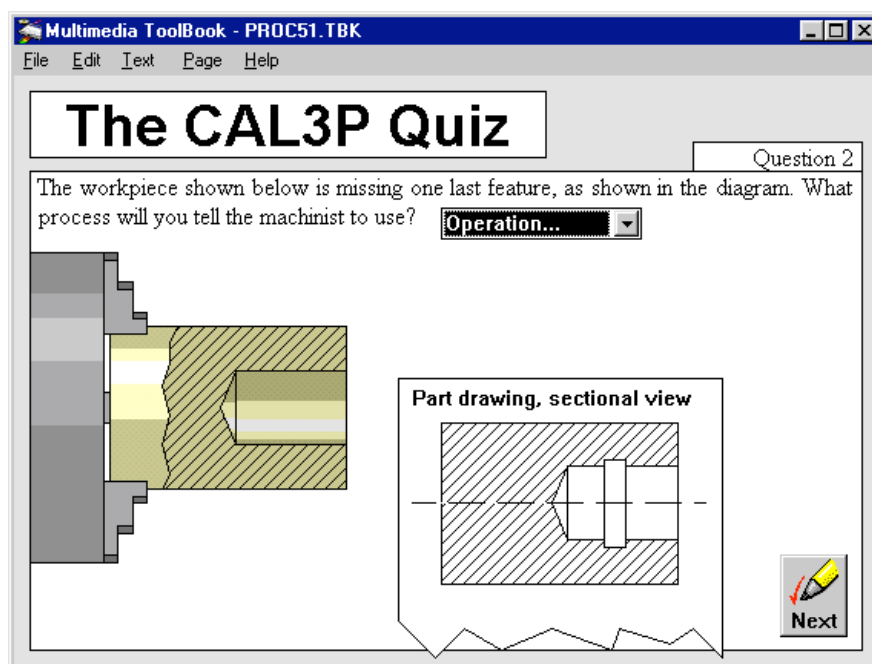


Figure 7.6: CAL3P quiz, question 2

7.3.3 Question 3: Lathe Nomenclature

Question 3 tested the student's knowledge of correct names for the different components of a typical lathe. A screen supplying this information was available within the informative section of CAL3P, to be visited at the student's discretion. Students would also have seen references to the chuck (and to a lesser extent the tailstock) while selecting operations for their own process plans.

The question presents an exact copy of the diagram previously made available to the student, but the pop-up information fields about the different parts of the lathe have been

replaced with three multiple choice questions (figure 7.7). The student was required to supply an answer to each.

Question 3a concerns the chuck, to be selected from a list of possible names; motor, clamp, chuck, jaws, grip. Question 3b concerns the saddle, the list of possible answers including vernier, saddle, slider, carousel, cross slide. Question 3c requires a name for the tailstock, selected from a list containing tool post, centre, steady, slider, tailstock.

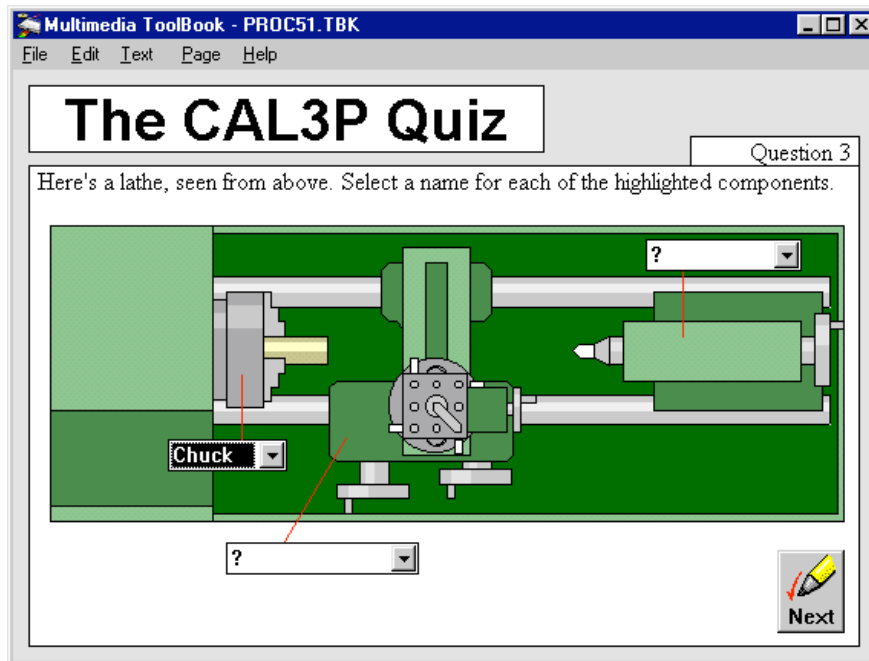


Figure 7.7: CAL3P quiz, question 3

7.3.4 Question 4: Turning Problems

Question 4 further investigated whether students visit 'tangential' information when it is offered. A section labelled 'A Guide to Turning Principles' had been offered to the students when they completed the informative section on the work of the process planner and were about to write a process plan of their own. Several pages into the section on workpiece deflection was a screen with a diagram which corresponded to the problem shown in this question (figure 7.8).

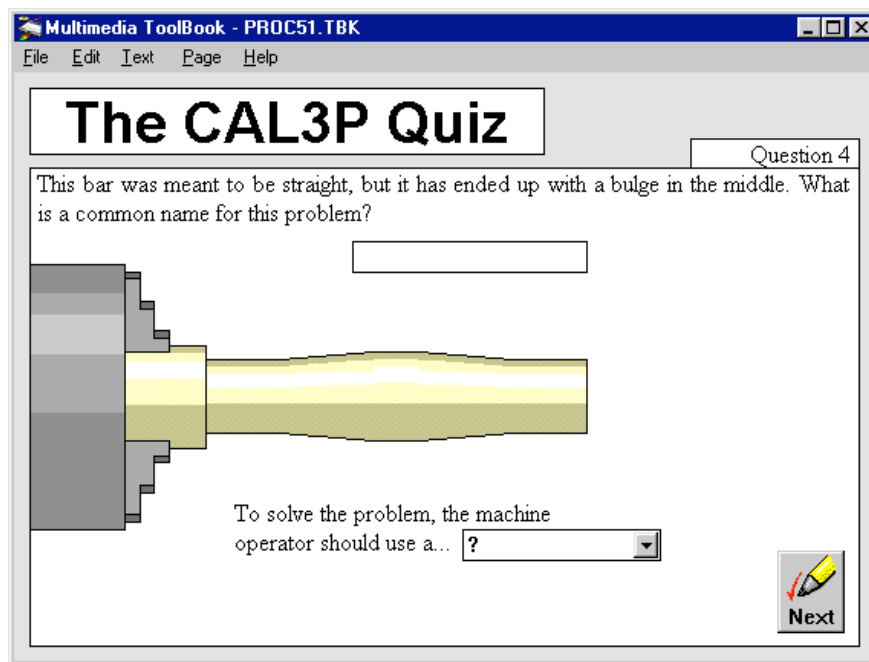


Figure 7.8: CAL3P quiz, question 4

First, the student was required to name the problem. A blank text box with a flashing cursor awaited the student's input, which should be to identify the problem as 'barrelling'. Since this must be done from memory and not simply picked from a multiple choice list, some alternative spellings were also accepted as correct answers.

Question 4b is a multiple choice question where the student is required to name something that should be used to prevent the barrelling problem. Possible answers are sharper tool, steady, deeper cut, centre support, 4 jaw chuck. The correct answer is that a steady should be used. For deflection to result in a part shaped like a barrel, a centre support would already be in use.

7.3.5 Question 5: Turning Between Centres

The pair of questions on page 5 of the quiz were specifically meant to identify students who had prior experience with machine tools; answering these questions correctly required information not given within CAL3P. A diagram shows a workpiece mounted for turning between centres, but with no means of making the workpiece rotate (figure 7.9). It is necessary for the machinist to use a drive accessory to achieve this, and question 5a requires that the student name this. Common terms for this include drive dog and drive attachment, so any answer which included 'drive' was marked as correct. In use, it was found that some experienced engineers wanted to use the term dog and carrier,

and in one instance, driving dog so these answers were scored as correct and the software was updated to mark them as valid answers in the future.

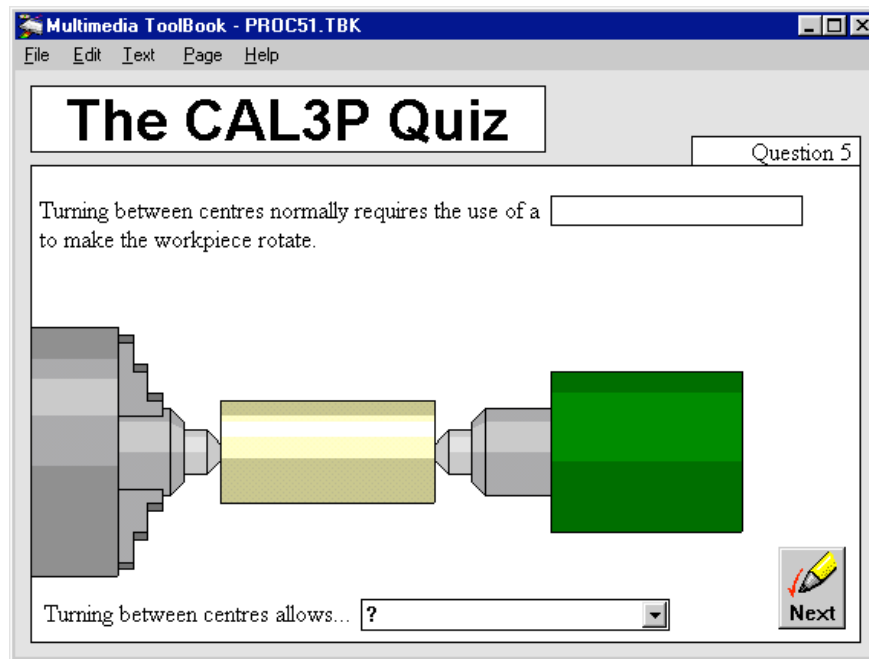


Figure 7.9: CAL3P quiz, question 5

Question 5b is a multiple choice pop-up, requiring that the student states what turning between centres might allow. Possible answers are faster cutting speeds, greater torque, longer tool life, better geometric accuracy, smoother surface finishes. The correct answer is that turning between centres allows better geometric accuracy.

Although some provision has been made in the CAL3P software for a future capability to represent mounting the workpiece between centres, the DCG code of the simulation module does not currently support this operation. A schematic diagram of the setup and some explanatory text exist, but the button which would allow turning between centres to be selected from the operations menu was set to 'not available', and no exercises were set which specifically required turning between centres. This question tests the student's existing knowledge of turning principles, not what they have learned from CAL3P.

It was anticipated that a correct answer here would imply that the student was an experienced engineer, and therefore that correct answers given to other quiz questions may not have come from knowledge gained during the CAL3P exercise. However, many students who should be considered novice engineers answered this question correctly. That turning between centres improves accuracy appears to have been common

knowledge; the level of prior knowledge among novices was higher than had been assumed. It may also be true that the alternative answers in this multiple choice question were too easily recognised as nonsensical.

7.3.6 Question 6: A Simple Turning Operation

The CAL3P exercise required students to specify a number of operations, by name and parameters. It was seen during observation of individual users that few could reliably enter instructions at first, but that an improvement took place during an hour with the software. This question assesses the student's ability to get a process planning instruction right first time, shortly after completing the planning exercise.

A part is shown, fully dimensioned, and the student must choose the operation name from a multiple choice pop-up, and then supply values for X, L and D in blank text boxes (figure 7.10).

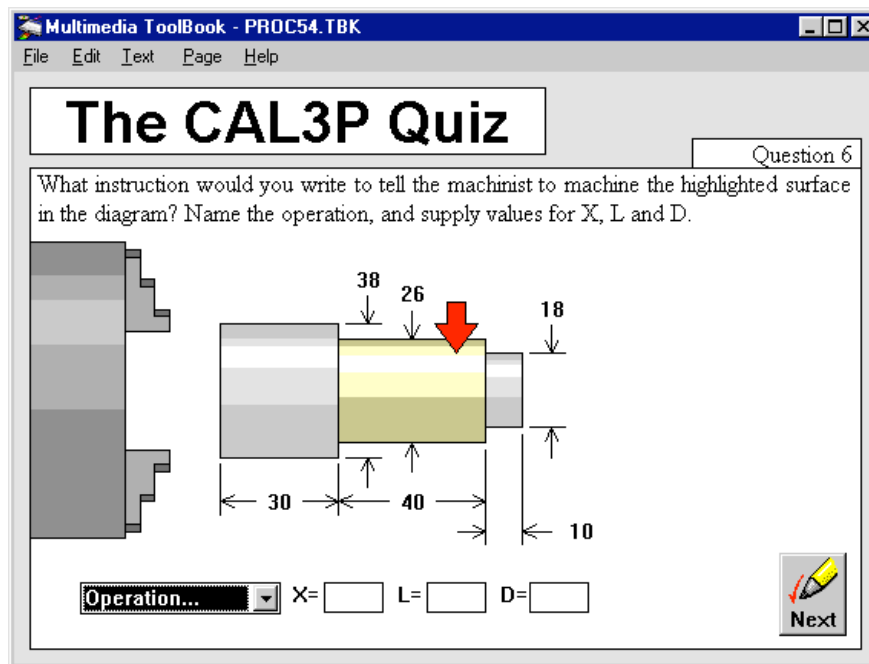


Figure 7.10: CAL3P quiz, question 6

Question 6a, specifying the name for the operation included the possible answers shoulder facing, straight turning, milling, end facing, knurling. Straight turning was the answer sought, and students would have recently specified a number of these instructions in their process planning exercise.

Question 6b required a value for X, which students had earlier been told is the distance 'down the bar' from its end before a cut starts. The correct value for X in this

question was 10. Question 6c required a value for L , which students had been told is the length of the cutting operation. The correct value for L was 40. Question 6d required a value for D , which students had used to specify the final diameter of a cylindrical feature after cutting. In this question, the correct value for D was 20.

7.3.7 Question 7: Parting Off

Question 7 assesses the student's ability to make use of diagrammatic and textual descriptions of manufacturing operations when confronted by a real world situation. This application of knowledge gained is the phase of the learning process which is known as 'transfer'.

Question 7 simply featured a video window and 'play' button, and the question 'What is the name of this operation?' (figure 7.11). A blank text box with a flashing cursor awaited their answer.

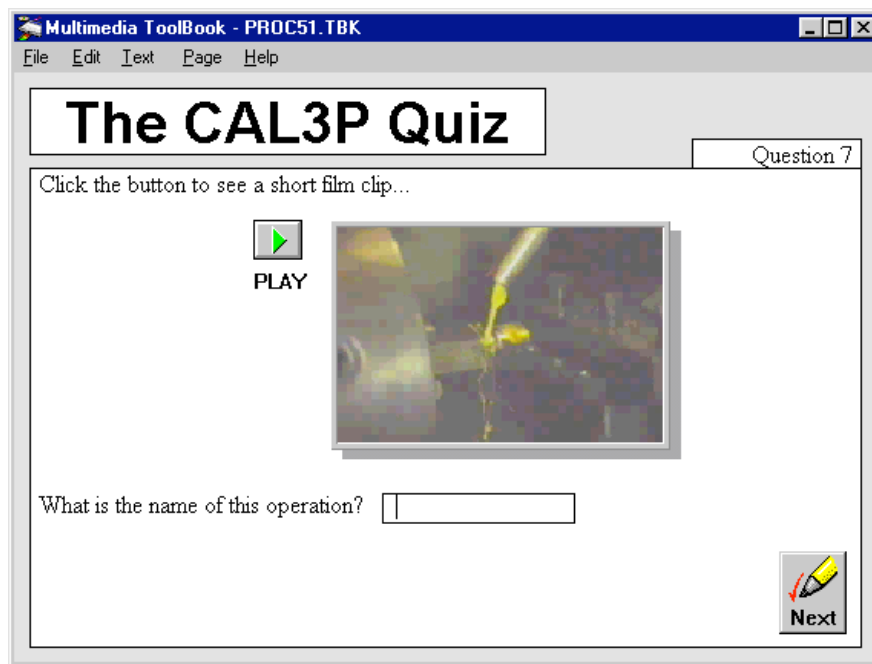


Figure 7.11: CAL3P quiz, question 7

The students had not been able to see this video clip elsewhere within the informative sections of the software, but the process plan they created during the exercise may well have included a parting off instruction at some point. Unfortunately, problems with the way the machines were set up in the computer lab where the CAL3P exercise was run meant that it was not possible to display video clips during some of the trials, so experimental data for this question are not always available. Also, none of the computers

used by students was equipped with speakers, so the noises of machining processes taking place could not be heard.

7.3.8 Question 8: The 4-jaw Chuck

In a further effort to identify students with previous knowledge of engineering, this question tested for information not supplied within CAL3P. A photo of a four jaw chuck was shown, and the student was asked 'Why might the machinist need to use one of these instead of a conventional three jaw type?' (figure 7.12).

Possible answers from the multiple choice list were for holding larger workpieces, for increased safety, for gripping workpieces which aren't round, for gripping softer materials, for machining off-centre features. The answer sought by the software is that the 4-jaw chuck allows the machining of off-centre features.

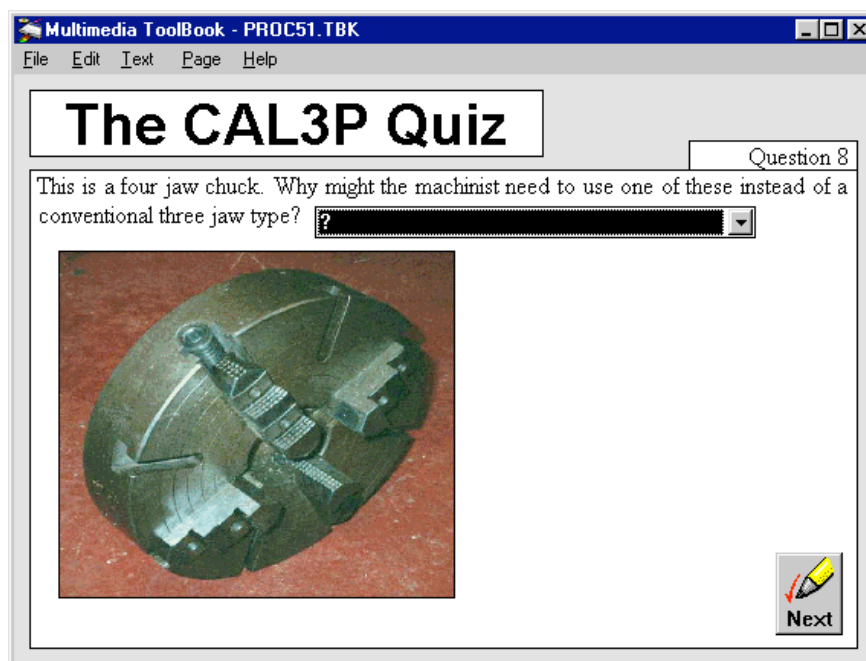


Figure 7.12: CAL3P quiz, question 8

This question gave rise to some protest, because many students had selected the answer 'gripping workpieces which aren't round.' A four jaw chuck is often used for this purpose, but it is not universally applicable to non-round workpieces. Hexagonal or triangular bar, for example, is better held in a three-jaw chuck, and it will sometimes be necessary to use a four jaw chuck to grip a round workpiece, held such that it is off centre, allowing subsequent machining operations to produce an offset feature.

7.3.9 Question 9: Influences on the Process Planner

Question 9 tested the student's knowledge of the information which must be taken into account by the process planner. This had been covered in the introductory lecture, and was also addressed in the informative section of the software.

Six buttons were shown on screen, which could be dragged into position on a list, ranking from 'very important' to 'not important' (figure 7.13). It was hoped that the student would remember the key issues and select part drawings, estimated demand and factory capabilities. In truth, the order of these three is not tested, but they should all feature in the list before the other three issues, which the students had not been told were important to the process planner; sales price, product lifetime and environmental issues.

The question is marked in three parts, with 9a testing for 'part drawings', 9b testing for 'estimated demand' and 9c testing for 'factory capabilities', each of which were marked as correct if they appeared somewhere in the top three.

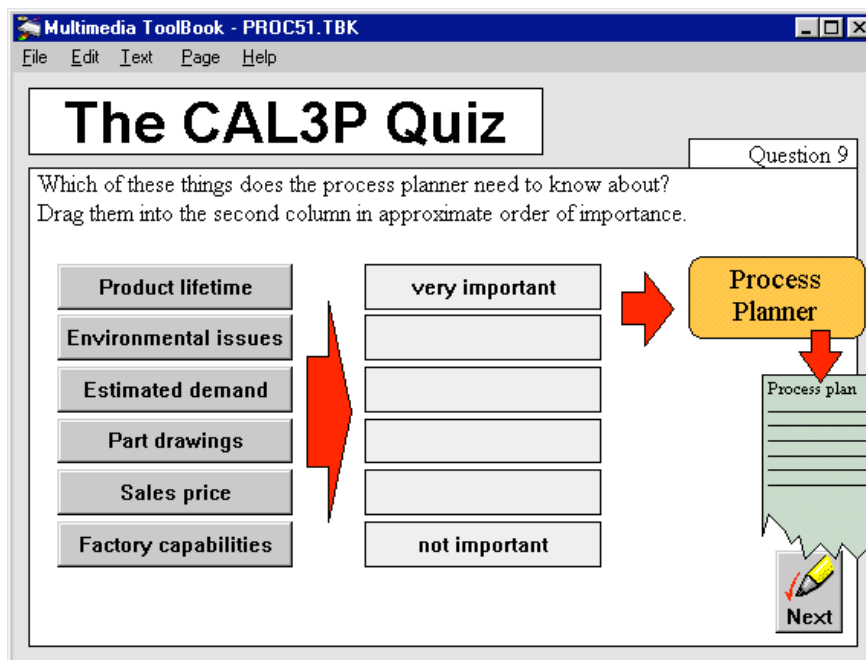


Figure 7.13: CAL3P quiz, question 9

7.3.10 Question 10: Taper Turning

The final question showed a workpiece gripped in the chuck and asked which instruction would be required to make an indicated feature (Figure 7.14). This is a tapered feature, and students should have already specified a taper turning instruction in their own process plans. The question is difficult because the possible answers given in the

pop-up list are quite similar. The student must visualise what each of the instructions would do and select the correct answer.

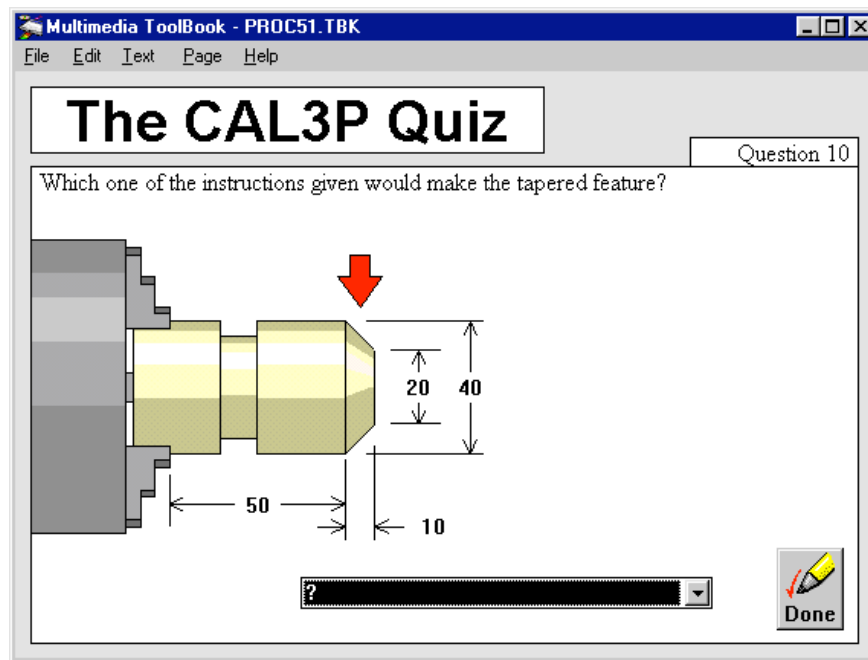


Figure 7.14: CAL3P quiz, question 10

The list of answers from which the student must choose includes:

Chamfer X=0 L=10 D1=20 D2=40

Taper turn X=10 L=10 A=45

Chamfer X=0 L=10 D1=40 A=45

Bar turn X=0 L=10 D=40 A=45

Taper turn X=0 L=10 D=20 A=45

The last answer is the correct one. Though a small tapered feature on the end of a cylindrical part is often called a chamfer, this is not the name of a manufacturing operation.

7.3.11 Presenting Quiz Results

After the ten pages of questions, the student reaches the summary page, which shows whether his answers are correct or not (Figure 7.15). This shows his responses as correct (green) or incorrect (red). The markers are also shaped differently in order to be intelligible to a colour-blind student. From this screen it is possible to click on one of the ten 'question mark' buttons to see an explanation of the correct answer to each of the questions. Figure 7.16 shows an example of one such page. Students were encouraged to

visit the pages corresponding to each incorrect answer. Finally, selecting the 'exit' button saved their results and ended the CAL3P session.

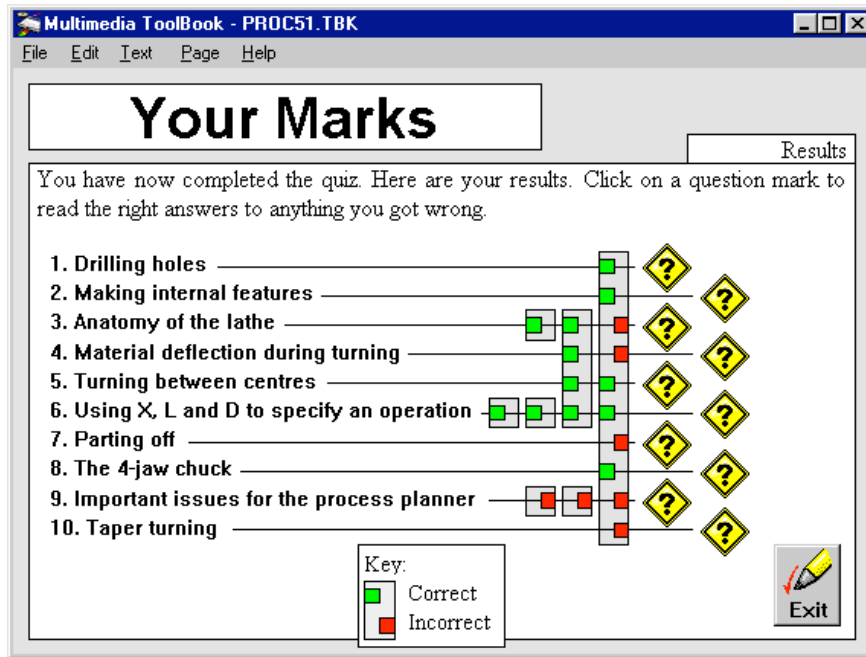


Figure 7.15: Student quiz performance display

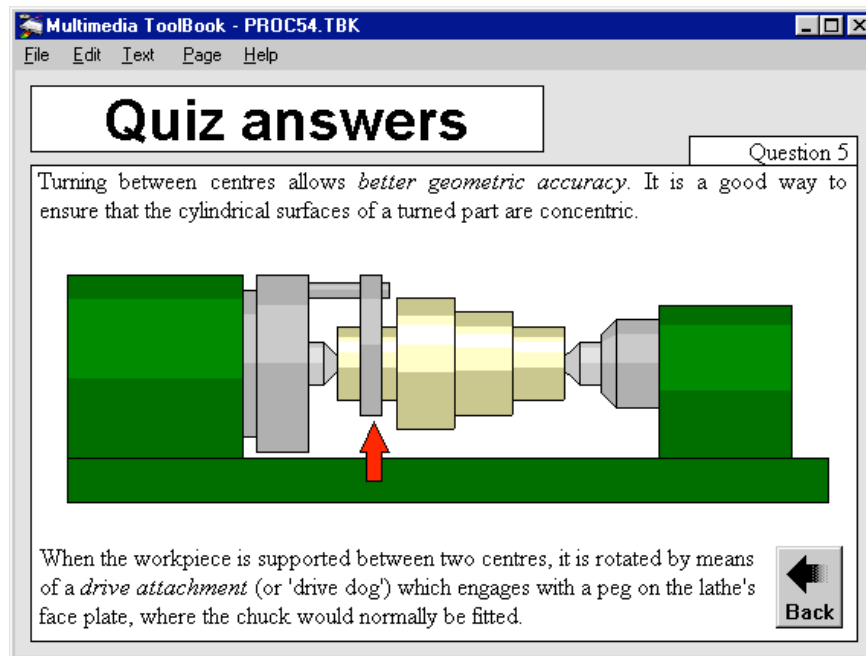


Figure 7.16: Sample screen explaining an answer demanded by the quiz

CAL3P's quiz questions are summarised in Table 7.1, showing the method in which the question is tested and the likelihood that the answer will have been found during the student exercise.

Question	Subject	Format	Method	Done in exercise?
1	Drilling	stage drawing	drag and drop	definitely
2	Internal bore	part drawing	list box	optional
3(a)	Lathe parts - chuck	graphic	list box	definitely
(b)	- saddle		list box	optional
(c)	- tailstock		list box	definitely
4(a)	Barrelling	stage drawing	missing word	optional
(b)	- use a steady		list box	no
5(a)	Turn between centres	stage drawing	missing word	no
(b)	- better accuracy		list box	no
6(a)	Straight turn	part drawing	list box	definitely
(b)	X value		missing value	definitely
(c)	L value		missing value	definitely
(d)	D value		missing value	definitely
7	Part off	movie	missing word	plan-dependent
8	Four jaw chuck	photograph	list box	no
9(a)	Part drawings	text	drag and drop	definitely
(b)	Volume		drag and drop	definitely
(c)	Factory capabilities		drag and drop	definitely
10	Taper turn	stage drawing	list box	definitely

Table 7.1: Quiz question summary

7.4 Results Obtained from the Quiz

Four sets of experimental data were gathered with the on-line quiz; participants were a group of M.Sc. students, two groups of second year undergraduates studying for B.Eng. degrees in Manufacturing Management and Manufacturing Engineering, and a small group of part-time students from Bolton Institute of Higher Education studying B.Sc. Manufacturing Systems Design.

Experimental results are presented in Tables 7.2 to 7.5, showing the performance of each student in the 19 questions that make up the quiz. Each student was also given a subjective mark for their process plan by the author. This was based on the final geometry of the workpiece and the safety and practicality of their work instructions, found by carrying out a CAL3P machining simulation using the student's process plan; from a maximum score of ten points, marks were deducted for each incorrect feature, for failing to face off the bar, or for gripping the bar in an unrealistic manner. In some cases students

failed to save their process plans before starting the quiz, so this information is not available; 'N/S' (not stored) is entered in place of a grade for their work. Students are identified by their initials.

Question	Student results											
	ST	PS	LG	JK	HK	DF	CH	TB	MT	FR	AB	JD
1	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓
2	✓	✗	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓
3 (a)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
(b)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗
(c)	✓	✓	✓	✗	✓	✗	✓	✓	✗	✗	✓	✗
4 (a)	✗	✗	✓	✗	✓	✗	✓	✓	✗	✗	✓	✗
(b)	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
5 (a)	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
(b)	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓
6 (a)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
(b)	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
(c)	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
(d)	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	✓
8	★	★	★	✓	★	✓	✓	★	★	✗	★	✓
9 (a)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(b)	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	✓	✓
(c)	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓
10	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✓	✓
Plan mark	9	9	7	7	8	N/S	9	N/S	7	5	8	8

Table 7.2: CAL3P quiz results, M.Sc. students

In each table, the symbol '★' identifies students whose answer to Question 8 deserves half a mark, as explained in Section 7.3.8; their answer is to some extent correct, but a better one was available.

From the M.Sc. group's answers (Table 7.2) to CAL3P's 'trick questions' it was deduced that HK was an experienced engineer, leaving 11 students whose results could be used to indicate where the most effective learning had taken place during the session. Experienced engineers were identified as being any person or group who answered at least three of the four 'trick questions' (4b, 5a, 5b and 8) correctly.

Question	Student results											
	TK	AS	YM	PB	JR	DB	AL	AM	BC	LM	RY	MS
1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
2	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✗
3 (a)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗
(b)	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗
(c)	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✓	✗
4 (a)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
(b)	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
5 (a)	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗
(b)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
6 (a)	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
(b)	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓
(c)	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
(d)	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
7	N/T	N/T	N/T	N/T	N/T	N/T	N/T	N/T	N/T	N/T	N/T	N/T
8	✓	✗	★	✗	✗	★	★	★	✗	★	✓	✗
9 (a)	✗	✓	✗	✓	✓	✓	✗	✗	✓	✓	✓	✓
(b)	✓	✓	✗	✓	✓	✓	✓	✓	✗	✗	✓	✓
(c)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓	✗	✗
Plan mark	10	9	9	8	8	9	9	9	6	7	9	7

Table 7.3: CAL3P quiz results, Manufacturing Management course

Among the Manufacturing Managers (Table 7.3), TK and AL were identified as experienced engineers, leaving ten students who could be considered when assessing the amount of learning that had taken place.

Question	Student results								
	CM	JT	JP	L&BC&K	AG	N&N	ER	DE	
1	✓	✓	✓	✓	✓	✓	✗	✓	✓
2	✗	✓	✓	✓	✓	✗	✓	✓	✓
3 (a)	✓	✓	✓	✓	✓	✗	✓	✓	✓
3 (b)	✗	✓	✓	✓	✗	✗	✓	✓	✓
3 (c)	✓	✓	✓	✓	✓	✗	✓	✓	✓
4 (a)	✓	✗	✗	✓	✓	✗	✗	✗	✗
4 (b)	✗	✓	✗	✓	✗	✗	✗	✗	✓
5 (a)	✗	✗	✗	✓	✗	✗	✗	✗	✓
5 (b)	✓	✗	✓	✓	✓	✗	✓	✗	✓
6 (a)	✓	✓	✓	✓	✓	✓	✓	✓	✓
6 (b)	✓	✓	✓	✓	✓	✓	✓	✗	✓
6 (c)	✓	✓	✓	✓	✓	✓	✓	✓	✓
6 (d)	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	N/T	N/T	N/T	N/T	N/T	N/T	N/T	N/T	N/T
8	★	★	★	✓	★	★	★	★	★
9 (a)	✓	✓	✓	✓	✓	✓	✓	✓	✓
9 (b)	✓	✓	✓	✓	✓	✓	✗	✓	✓
9 (c)	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	✓	✓	✗	✗	✗	✓	✗	✓	✓
Plan mark	10	9	9	8	8	9	9	9	6

Table 7.4: CAL3P quiz results, Manufacturing Engineering course

Among the Manufacturing Engineers (Table 7.4), where a pair of students were grouped together due to a shortage of computers this is represented with the ampersand symbol '&'. Students L&B and DE can be seen to be the most experienced engineers among this group, leaving seven sets of experimental data available for use. Among the B.I.H.E. students (Table 7.5), CG appears to be an experienced engineer, leaving four sets of experimental data available.

A total of six sets of results were identified as being from students who should be considered experienced engineers. These were discounted from CAL3P's validation process. These students' results are summarised in Table 7.6. It can be seen that most of these students also scored very well on the other questions in the CAL3P quiz. While this cannot be taken as proof that they learned the answers during the session, it does at least offer evidence in support of the method used to identify experienced engineers; their performance on other questions is better than any of the class groups with a mean 81.3% of questions answered correctly. Further corroboration may be found in the marks awarded for the quality of their process plans; the mean mark among experienced engineers was 9.4, compared with 7.83 for novice engineers.

Question	Student results				
	CG	GR	DS	SB	MD
1	✓	✓	✓	✗	✓
2	✓	✗	✓	✗	✗
3 (a)	✓	✓	✓	✓	✓
(b)	✓	✓	✗	✗	✗
(c)	✓	✓	✓	✓	✓
4 (a)	✗	✗	✓	✗	✗
(b)	✓	✗	✗	✗	✗
5 (a)	✓	✗	✗	✗	✗
(b)	✓	✓	✓	✓	✓
6 (a)	✗	✓	✓	✓	✓
(b)	✓	✓	✗	✓	✓
(c)	✓	✓	✓	✗	✓
(d)	✓	✓	✓	✗	✓
7	✓	✗	✓	✗	✓
8	✓	★	✓	★	★
9 (a)	✓	✓	✓	✓	✓
(b)	✓	✓	✓	✓	✓
(c)	✓	✓	✓	✓	✓
10	✗	✓	✗	✓	✗
Plan mark	N/S	9	N/S	8	8

Table 7.5: CAL3P quiz results, Manufacturing Systems Design course at B.I.H.E.

Question	Student & course					
	HK MSc	TK MM	AL MM	L&B ME	DE ME	CG MSD
4 (b)	✓	✗	✓	✓	✓	✓
5 (a)	✓	✓	✓	✓	✓	✓
(b)	✓	✓	✓	✓	✓	✓
8	★	✓	★	✓	★	✓
Other questions	12	10*	10*	13*	13*	12
Plan mark	8	10	9	10	10	N/S

Table 7.6: Quiz performance by those identified as experienced engineers

* Mark out of 14 remaining questions, not 15 (video based question not tested)

Further investigation into these students' backgrounds confirmed that they had begun the CAL3P exercise with prior knowledge of machining. For instance, L&B were mature students with prior machining experience, DE was a university laboratory technician studying part-time for a further qualification, and CG was a part-time student whose other work involved the use of machine tools. This evidence supports the approach used.

For the remaining ‘novice engineer’ students, each of the questions in the quiz was considered. Where a substantial number of novice engineers were seen to have answered a question correctly, it was assumed that this was the result of learning which had taken place during the CAL3P session. With six students or pairs identified as experienced engineers, 32 students or pairs remained. Each question (or part-question) was answered either correctly or incorrectly, giving a total out of 32 from which the effectiveness of the IEM technique could be inferred. In the next chapter, Section 8.9 discusses the results obtained.

7.4.1 Results from CAL3P’s ‘trick questions’

Questions 4(b), 5(a) and 5(b) are some of CAL3P’s ‘trick questions’, meant to identify experienced engineers. Of those defined as novice engineers, only 1 student answered Question 4(b) correctly. Similarly, only 2 students answered Question 5(a) correctly. Since Question 5(b) was correctly answered by 22 novice engineers, the answer to this question appears to be fairly common knowledge. As such, it was not as useful for identifying experienced engineers as had been hoped. The final ‘trick question’, Question 8 was answered correctly by 6 novice engineers, though the response ‘for gripping workpieces which aren’t round’ is a somewhat valid answer (see Section 7.3.8), and a further 20 novices gave this answer.

7.5 Student Evaluation of CAL3P

In addition to measuring the performance of the students, it was considered necessary to record their opinions, to discover if they enjoyed CAL3P and hopefully to discover which media they believed were most worthwhile. After the quiz, students were given an evaluation form similar to that which had been used with the prototype software, though the list of questions was expanded to include questions meant to assess the students’ reaction to media such as photographs. In the nine questions asked of the students, numbers 1, 3, 4, 5 and 7 are analogous to questions asked of the M.Sc. students who used the HyperCard prototype of CAL3P (see section 6.3.2). The assessment form is reproduced below (figure 7.17):

Consider the following statements, and circle one number where 1 = strongly agree, and 5 = strongly disagree.

	(yes)			(no)	
	1	2	3	4	5
1. The way to use the program was easy to understand.	1	2	3	4	5
2. CAL3P made writing a process plan easy.	1	2	3	4	5
3. The CAL3P exercise was more enjoyable than a normal tutorial.	1	2	3	4	5
4. It was easy to understand the syntax required for a process plan.	1	2	3	4	5
5. The program adequately supports the theory we had learned.	1	2	3	4	5
6. The photos in CAL3P helped us understand turning principles.	1	2	3	4	5
7. The CAL3P exercise was more worthwhile than a normal lesson.	1	2	3	4	5
8. The diagrams in CAL3P helped to understand turning principles.	1	2	3	4	5
9. We were given plenty of time to use CAL3P properly.	1	2	3	4	5

Figure 7.17: Users' assessment form

As before, students were also invited to identify CAL3P's best and worst features, and identify anything else they had expected to see within the software. The set of questions presented here was used in all trials at the University of Salford from April 1997 onwards. Table 7.7 shows the mean responses from each group. Because the evaluation forms were submitted anonymously, it is not possible to make a distinction between the comments of experienced engineers and novices.

Statement	Course group		
	MSc	ME	MM
1	1.75	1.08	1.50
2	1.50	1.42	2.08
3	1.75	1.17	1.75
4	1.33	1.67	2.50
5	1.75	2.08	2.42
6	2.00	2.08	3.42
7	1.58	1.50	2.08
8	1.75	1.42	2.00
9	2.16	1.58	1.92

Table 7.7: Mean student evaluations of CAL3P (low numbers indicate agreement)

Higher scores indicate disagreement with the positive statements made on the evaluation form, and highlight shortcomings of the software. It can be seen that the general student reaction was favourable. Only one response from one group indicated a general disagreement, that being the Manufacturing Management group's evaluation of the usefulness of photographs within the software.

Responses to the nine statements allowed the quality of the exercise to be measured in terms of ease of use, relevance, the value of certain media and the pace of delivery. Each of these are discussed in the subsections that follow.

7.5.1 Ease of Use

For the first statement, "The way to use the program was easy to understand," scores of 1.75, 1.08 and 1.50 represented a substantial improvement over the earlier students' reaction to the first Prototype, which received a mark of 3.83 for a similar question (see Section 6.3.2).

A significant part of the learning process under CAL3P is the activity where students write a process plan of their own. Throughout development it was considered crucial that CAL3P allowed the user to concentrate on learning, rather than on the operation of the software itself. Learning theory implied that short-term memory was limited to the concurrent retention and manipulation of a very small number of facts, for a very short period of time (see Section 3.2.4). Thus, a student would struggle to enter machining instructions if operating the software required a great deal of attention or produced delays. The response to the second statement, "CAL3P made writing a process plan easy," shows that this goal has largely been achieved, with mean marks of 1.50, 1.42 and 2.08 from the respective groups.

The fourth statement, "It was easy to understand the syntax required for a process plan," received marks of 1.33, 1.67 and 2.50 from the three groups. Specifying areas of material to be removed during an operation using values for X, L and D (see Section 6.2.1) will have been unfamiliar to most if not all students before the exercise, but it was necessary to use an unambiguous format which could easily be interpreted by the computer (or in the workplace, a machinist). In requiring responses to this statement, any difficulties reported in 'ease of use' could be separated into those relating to the software interface and those relating to the syntax involved. Table 7.8 shows student responses to statements 1, 2 and 4 by the Manufacturing Management students, who were the most critical of the software's ease of use.

Attribute	Individual students' responses (Manufacturing Management)											Total	
	1 user interface	1	1	1	1	1	3	2	1	2	1		2
2 writing a plan	1	1	1	1	1	5	4	2	2	3	2	2	25
4 planning syntax	3	1	2	2	3	2	4	3	3	3	3	1	30

Table 7.8: Student evaluations for user interface, ease of use and syntax

No pattern is immediately apparent, though clearly the syntax itself was regarded as the most difficult aspect of operating the software with a total of 30. It is surprising that the student who most disliked the user interface found the planning syntax easier than most of his or her colleagues. Discovering how to specify operations in the right syntax was a vital part of the learning process, and as such it could not be simplified. Conversely, the user interface was meant to be clear from the outset, allowing students to navigate easily within the information available. The score of 18 for the user interface suggests this was largely achieved. The business of entering a process plan should be equally easy; the score of 25 is somewhat disappointing. This aspect of the software requires further work. Typically, the problems which were observed in writing a plan occurred when a student wanted to insert an additional instruction in the middle of an existing process plan.

7.5.2 An Enjoyable Exercise?

Marks of 1.75, 1.17 and 1.75 for the third statement, "The CAL3P exercise was more enjoyable than a normal tutorial," suggest that most students enjoyed the session. A similar reaction had been found with the prototype, but the extent of the students' positive reaction was not anticipated. Cann [1997] invited students to classify themselves with regard to their view of CAL based learning material. Approximately equal proportions of the students liked CAL, would use CAL when necessary, and disliked CAL. They named themselves the 'nerds', 'normals' and 'phobes' respectively. For the CAL3P trial, it was anticipated that a similar distribution of students would be encountered, with some students agreeing that using CAL3P was more enjoyable than a normal tutorial, some disagreeing and others indifferent. In fact, as the detailed breakdown of student responses to this question shows, the majority of students enjoyed using CAL3P (Table 7.9):

Group	Individual students' responses											
MSc	3	1	1	4	1	3	1	2	1	2	1	1
ME	2	1	1	1	1	1	1	1	2	1	1	1
MM	2	2	1	2	3	3	3	1	1	1	1	1

Table 7.9: Student responses to the statement “The CAL3P exercise was more enjoyable than a normal tutorial.” (1 = strongly agree, 5 = strongly disagree)

If Cann’s students’ classifications were applied to the students who gave feedback on CAL3P, a mark of 1 might be taken to indicate a ‘nerd’, 3 a ‘normal’ and 5 a ‘phobe’. It can be seen that the distribution of responses here does not match Cann’s, with 30 of the 36 students supporting the suggestion that using CAL3P was enjoyable, 23 of them expressing this strongly. Five were indifferent and one did not enjoy the lesson.

It may be that responses vary with the aptitude of the students themselves; the people who evaluated CAL3P were all accustomed to operating computers as an integral part of their engineering courses, while Cann’s students were from the Faculty of Medicine. Equally, the apparent quality of the software which the students are required to use will be an influencing factor – one which it was hoped to quantify in the course of the trials conducted.

Although, as learning theory suggests, a learning process is more effective if it is perceived to be enjoyable (e.g. Honigsbaum [1992]), it remained necessary to find out if the lessons learned were also thought of as relevant, or if CAL3P was simply an attractive diversion. Responses to the fifth statement, “The program adequately supports the theory we had learned,” were generally favourable with means of 1.75, 2.08 and 2.42 from the three groups. To the seventh statement, “The CAL3P exercise was more worthwhile than a normal lesson,” marks with a mean of 1.58, 1.50 and 2.08 were given. These results suggest agreement from most of the students.

7.5.3 Media Displayed by CAL3P

The sixth statement, “The photos in CAL3P helped us understand turning principles,” received the worst overall response from each group, scoring 2.00, 2.08 and 3.42. The mark of 3.42 was the only overall disagreement from a group during the trial. This must be considered in conjunction with responses to the eighth statement, “The diagrams in CAL3P helped to understand turning principles.” The diagrams fared better than the photographs, at 1.75, 1.42 and 2.00. CAL3P contains eight photographs, typically located on pages where individual machining operations are specified, showing the corresponding equipment which will be used. It may be significant that the photographs in CAL3P are only shown on a pop-up basis rather than being used as the standard

method of communication with the user. Some students may not have bothered to view them all.

7.5.4 CAL3P Exercise Duration

This final statement, "We were given plenty of time to use CAL3P properly," elicited some unusual responses in that some students disagreed despite the fact that they were not operating under a time constraint. Only the M.Sc. students who used CAL3P in December 1997 were required to finish by a set time. Still, in some cases, students in other groups responded to this question with a 2, 3 or even 4. Mean responses were 2.16, 1.58 and 1.92 from the M.Sc., ME and MM groups respectively. Correlating individual responses to those made in reaction to other statements, it appears that some students who stated they had found the exercise enjoyable and worthwhile considered the time spent on CAL3P too short, despite the fact that they had completed the required exercise. This may represent another endorsement of the software.

7.5.5 Student Evaluation at B.I.H.E.

At Bolton Institute, students who worked through the same process planning exercise were given an evaluation sheet designed by their course tutor. Many of the questions are analogous to those used in the rest of the trials. Students were required to respond to a number of questions by marking a point on a scale, as shown in the example presented in Figure 7.18:

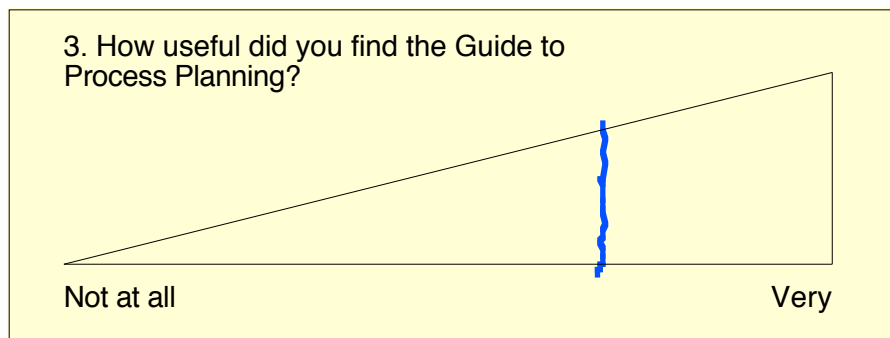


Figure 7.18: Student evaluation method employed at B.I.H.E.

There were six questions of this kind. It was a simple matter to measure the point along the 100mm baseline of the triangle to the point at which the students had made their marks. In this way, the students' responses could be quantified as percentages. The questions, and the scales on which they were measured are summarised below (table 7.10).

Question	Answer scale
1 Are you an experienced process planner? Indicate your level of experience.	(none – fully skilled)
2 How useful did you find the instructions?	(not at all – very)
3 How useful did you find the Guide to Process Planning?	(not at all – very)
4 Did you find writing the process plan easy or hard after using the guide?	(easy – hard)
5 Did you find the process simulation useful?	(not at all – very)
6 Did you find the quiz useful?	(not at all – very)
7 Did you find the package as a whole useful as an introduction to process planning?	(zero – 100%)
8 In your opinion should this be developed into a process planning aid for all processes?	(no – yes)

Table 7.10: Student evaluation questions set at B.I.H.E.

Table 7.11 shows the students’ responses:

Question	Student responses				
	CG	GR	DS	SB	MD
1	73	7	36	32	55
2	57	63	90	60	68
3	72	64	67	76	63
4	43	32	79	66	42
5	50	86	89	97	72
6	22	38	59	74	45
7	44	74	92	94	60
8	22	100	–	91	47

Table 7.11: Student responses to CAL3P at B.I.H.E.

(‘–’ indicates no response given)

Since the assessment forms used at B.I.H.E. were not submitted anonymously, it was possible to look for a pattern between student performance in the quiz and their reactions to the exercise.

A correlation can be found between the students’ assessment of their existing skill level (required by the first question) and their performance against the sections of CAL3P’s quiz which were designed to identify experienced engineers. Table 7.12 shows how the student CG’s assessment of his skill supports his having been identified as an experienced engineer for the purposes of CAL3P validation.

Question	Student				
	CG	GR	DS	SB	MD
4 (b)	✓	✗	✗	✗	✗
5 (a)	✓	✗	✗	✗	✗
5 (b)	✓	✓	✓	✓	✓
8	✓	★	✓	★	★
Student's skill assessment	73	7	36	32	55

Table 7.12: Performance among B.I.H.E. students on CAL3P's 'trick questions'

Predictably, CG's response to the question "Did you find the package as a whole useful as an introduction to process planning?" received a less enthusiastic response than from his colleagues who had more to learn from the software. The other questions in these students' evaluation provide an insight into the perceived value of the informative section 'A Guide to Process Planning', the DCG simulations and the quiz. Students found the simulations most valuable, giving a mean mark of 78.8%. The guide to process planning, which offered much less interactivity received a mean of 68.4%, and the quiz (which is entirely linear) received only 47.6%.

The questions employed by the B.I.H.E. course tutor did not allow a direct comparison with those used in evaluation at the University of Salford in every case, but overall a favourable reaction to the software can be seen. It was interesting to discover that the session in which CAL3P was employed had taken a slightly different form at B.I.H.E.; after completing the quiz each student launched the software again and demonstrated their process plan to their classmates, who could then criticise the approach taken. CAL3P offered a useful learning exercise at B.I.H.E. even though it was employed in a way which had never been envisaged.

7.6 Student Comments

In addition to giving a numerical response to a set of questions, students were free to make other comments. As with the trials of the prototype (detailed in Section 6.3.2), students were invited to identify what they considered best and worst about the software, and to suggest any other features they had expected to see.

Students' comments are not directly quantifiable in terms of the success or failure of the software, but this invitation allowed students to contribute ideas which would not have emerged from their numeric responses alone. Some of the suggestions which had been made when the prototype was tested influenced the final version of the software

employed during these trials. Comments gathered at this stage give an insight into programming required to develop the software still further.

Not all students offered comment, but some made several observations. Where the comments made by different students were similar they have been grouped together in the lists below.

Statement	Occurrences
Easy to understand, easy to use	11
Enjoyable, more interesting than a normal lesson	5
Good visualisation, good simulations	4
Able to repeat problem areas, learn from mistakes	3
It was practical	3
Useful for novices / apprentices	3
Fast computer response time	2
Good layout	2
Built-in documentation useful	1
Being able to work alone	1
Gave an insight into a task never seen before	1
Prompts when specifying instructions	1
Enjoyed the quiz	1
Hope we can do more exercises like this	1
Good information on processes	1
Interactive learning preferred over lectures	1
“A worthwhile lesson”	1
“Best lesson in ages”	1
TOTAL	43

Table 7.13: Positive statements made about the CAL3P exercise

It is difficult to build up a complete picture of the success or failure of the software from comments such as these, but it is plain that the total number of positive comments far outweighed those which were negative. Of the negative comments made, many referred to factors external to the software itself such as the duration of the session or the nature of the introduction given. Of the remainder, some simple problems like a typographical error were easily remedied. Constructive criticism obtained by asking students to list any features they had expected to see in the software provided suggestions as to the direction future development might take. It would appear that more photographs are expected, a fact which would not have been clear from the students' evaluation of the photographs currently included in the software (see Section 7.5.3).

Statement	Occurrences
Took too long to find and rectify a problem	2
Didn't like the quiz / scored badly on the quiz	2
Out of practice interpreting engineering drawings	2
Needed more time	2
Question on 4 jaw chuck not covered in theory - unfair	2
Reported a typo or grammatical error	2
Needs to display all tools during simulation	1
Introduction too brief	1
Not very satisfied with the result	1
Wanted to work alone	1
Unable to resequence lines in the process plan	1
Not related to real environment	1
No video capability	1
Handling of syntax errors	1
Had no prior knowledge of machine tools	1
TOTAL	21

Table 7.14: Negative statements made about the CAL3P exercise

Statement	Occurrences
More pictures or photographs	4
Wanted dimensions on the finished workpiece	3
More help, step-by-step help	3
More explanation about other processes	2
Expected to have a demonstration first	2
More video clips	1
Generation of CNC code	1
Automatic calculation of speed and feed values	1
Ability to insert new lines in the middle of the process	1
TOTAL	18

Table 7.15: Features CAL3P users had expected to see

7.7 Critical Analysis

Section 7.5 presented the responses of the students who took part in the CAL3P software trials, showing that a clear majority had considered the exercise to be more enjoyable and worthwhile than a normal tutorial. Unfortunately, for the software to have been popular and to have made the learners feel confident were not the sole objectives of the program. Before CAL3P can reasonably be declared a success, several issues must be addressed:

-
- Could a similar quality of learning experience be offered by conventional means, at a lower cost in terms of resources and preparation time?
 - Does the built-in quiz adequately reflect the learning process which precedes it, and test students suitably?
 - Is permanent learning being facilitated, or does the testing process merely indicate that facts are being memorised in the short term?
 - Does the apparent motivation on the part of the learners come about as a result of the obvious novelty of the process planning tutorial, employing CAL when few if any of their other tutorial sessions do?

These possibilities are discussed in the subsections that follow.

7.7.1 IEM Development versus Improvements in Conventional Instruction

Even if it is accepted that CAL3P was seen to be effective, there remains a potential argument that the same learning outcomes might have been achieved more easily by conventional means. To test the assumption that multimedia was somehow bound to improve a teaching process, and in any event to quantify its impact and cost were key aims of this research, as listed in Section 2.1. As a result it is necessary to look beyond the apparently satisfied users. That an enjoyable, worthwhile computer-based exercise should arise out of several years' research and programming does not in itself go any way towards making the technology more accessible.

Viewed in this way, CAL3P was not economic to develop. It must be remembered, however, that the software is a piece of test apparatus. It was not devised simply to be an economic solution to the problem of giving novice engineers an appreciation of process planning, but to allow data to be gathered. Special features incorporated for this purpose made it more complex and thus expensive (in time at least) to develop – and actually inhibited its ability to help students learn somewhat, since the wide variety of styles and media within the software made the user interface harder to master. The duration of this project, then, and the complexity of the software produced are not indicative of what would be achieved under more normal circumstances where the developers are interested predominantly in computerising a topic, and not in research as such.

Any software design process begins by looking at the task that is to be computerised; in this case the teaching of process planning. When introducing a computer to a situation, it should not be an aim to use the machine to speed up a task which remains flawed. Stepping back and taking time to consider a task may not normally be an easy thing to do, given the demands of the work itself. As a result, it may be that the process has not been assessed for possible improvement for some time. Computerisation forces

those involved in the task to agree exactly what the task entails, and explain it to outsiders such as programmers. This consultation may result in suggestions which would not normally have emerged. (Thus, it could be argued, teaching of the subject is liable to be improved somewhat, regardless of whether it ultimately makes use of information technology or not.) This means that there exists a very real pitfall when claiming that CAL3P or any other IEM software has improved instruction; the standard of conventional teaching is also likely to improve while development takes place. When CAL3P was developed, new slides and student handout material were created for use in an introductory lecture on process planning. New student exercises were also developed as a direct result of the introduction of CAL3P. All of these must have contributed to the success observed, despite the fact that they are not features of the software as such.

As Section 6.2 described, however, the teaching of process planning by conventional means did not work at all well. Students with an inadequate grasp of engineering fundamentals were seen to struggle, and groups of students were slowed down, requiring regular feedback for the tutor who attempted to interpret their process plans and explain likely pitfalls with the aid of sketches. By supplementing the tutor with a set of computers which could perform the same task, a bottleneck was removed and the learning process was enhanced as a result. This should be a key principle in the selection of course elements to computerise; not to simply replace a set of activities, but to select those where the computer can actually add value to the learning process.

7.7.2 The Difficulty Level of the Quiz

It may be considered that some of the questions within the CAL3P quiz are too easy. However, it is often identified as an advantage of CAL software that it can cater for learners of differing ability, and the students who used CAL3P have been identified as such. The quiz questions offered 'something for everybody', particularly since most did not require significant prior knowledge.

It was aimed to use simply worded quiz questions such that students could if necessary demonstrate their understanding of the machining processes involved, even if their theoretical knowledge of the subject was scant. (In some cases it was necessary for the computer to show and name specific pieces of equipment as a part of the question. These were of course placed after the question in which students were required to identify some major parts of the lathe.) Not one of the nineteen questions in the quiz was answered correctly by every student; all contributed towards the experimental process.

Naturally, it is impossible to be certain that the question relating to, for example, straight turning was of the same level of difficulty as the question relating to parting off, or that relating to the influences upon the process planner. It is equally impossible to state

that each of the concepts taught was equally easy or difficult to grasp. To address this point, numerous screen displays from CAL3P have been included; Chapter 6 shows what the students experienced when using the software, Appendix 5 shows a model answer to the students process planning exercise and Section 7.3 shows the screens seen by a student when working through the quiz, also discussing the options listed in multiple choice questions. From this information it can be concluded that the level of difficulty involved in learning each principle and answering each question does not differ significantly. Consequently, students' performance in the quiz indicates the level of learning which occurred. This is explored the next chapter.

7.7.3 Deep Learning versus Surface Learning

One disadvantage which is often attributed to CAL quizzes is that with the test following so soon after the information has been supplied it is difficult to prove that 'deep' learning has taken place. The disadvantages of quizzes were described in Section 4.3.1.

In an effort to measure understanding rather than memory, most of the questions in the CAL3P quiz were made to test for understanding of principles, not simply the repetition of facts. Few of the quiz questions were presented in the same manner that the information containing the correct answer was originally supplied. For example, Question 6 demands that the student supply a manufacturing instruction to machine a feature; the student must determine the appropriate answer, not simply repeat a fact.

It is also significant that CAL3P does not use its built-in quiz as a barrier to further sections. In an effort to increase the level of interaction in a piece of IEM software, some packages include a short test after only a few minutes of reading. If the student fails to achieve the necessary pass mark, they have to repeat the section. As Section 4.3.1 identified, this leads to rote learning rather than any real effort to understand. When CAL3P is used, the quiz is separated from the theoretical element by almost an hour, during which the students write their own process plan and experiment with the manufacturing simulation. By the time they reach the quiz, it is most unlikely that the students will be able to recall the necessary answers unless they have actually gone through a successful process of attention, coding, storage and retrieval. Their new knowledge is still fragile of course, and subject to being forgotten (as described in Section 3.2.5); certainly, further rehearsal in the days to come would reinforce the learning that has taken place, but the same would still be necessary if the students had learned by conventional means.

7.7.4 The Source of Students' Motivation

Having received enthusiastic responses from almost all the students in the software trial, it became necessary to examine their source. As the reproduced evaluation forms in Section 7.5 and Section 7.5.5 show, the students were not coerced into these statements with leading questions or by being offered an undesirable alternative. They appear to have genuinely enjoyed using CAL3P. When invited to add their own comments in an unstructured section of the evaluation, their remarks confirmed this.

It remained possible that the students' opinion was swayed simply because they were doing something different, or perhaps because they were pleased to be invited to give feedback. The problem of results being influenced by subjects' awareness that a research study is taking place is sometimes referred to as the Hawthorne Effect. The Larousse Dictionary of Science and Technology describes it thus:

“The observation that experimental subjects who are aware that they are part of an experiment often perform better than totally naive subjects.”

[Walker, 1995]

The original Hawthorne Experiment involved varying a number of influences in a factory, such as the level of illumination. It was hoped to find optimum levels for workers' comfort and productivity. At each level of brightness, those consulted reported that they felt happiest. This shows how easily the experimental results obtained can be biased by the observation process. In the case of CAL3P's trials, the act of polling students may have produced an artificially good impression of their feelings towards the software. Of course, this same effect might also be precipitated when assessing the popularity of other CAL or IEM software, or other teaching methods.

To reduce the anticipated Hawthorne effect as far as possible, students who used CAL3P were not told that their process plans and quiz results were to be stored and later subjected to scrutiny. Also, the evaluation form was given to the students only at the end of the exercise, keeping them unaware that they were involved in a trial while they actually used the software. It should be noted that positive responses to CAL3P were not confined entirely to students who took part in the evaluation process; informal feedback during other demonstrations had also been encouraging. Of course, computer-based learning activities still have some novelty value for most people, so perhaps this component of their positive response is unavoidable. If it is inevitable that a novel, interactive learning experience of the kind that CAL3P offers will increase motivation, perhaps we should not probe into this too deeply, but rather make use of the phenomenon to make education more enjoyable and effective.

7.8 Conclusions

A method has been described whereby the effectiveness of learning facilitated by the CAL3P software could be measured. By storing students' answers to a specially designed quiz it was possible to assess the learning of individual facts taught by a variety of media. Programming the software to write quiz results to disk allowed trials to be conducted at a remote site with little logistic impediment. The built-in quiz reduced the tutor's workload because it was no longer necessary to mark a paper-based exercise, whilst increasing the educational value of the CAL3P session by offering rapid reinforcement.

Results were obtained from which the effectiveness of several distinct approaches to IEM could be inferred. Where quiz questions were answered correctly in a relatively high proportion of cases, it could be concluded that the teaching method employed was effective. In this way it was found that students performed best when answering questions which related to material they had encountered while entering process plans of their own. This was the expected result, that learning by doing is more effective than passive learning. Experimenting with CAL3P's machining simulation is an iterative process, in accordance with the cyclic model of learning presented in Chapter 3.

Information which was contained within sections accessed optionally via hypermedia links produced poor quiz results. It had been felt that allowing students to move to other sections by hyperlink would give them access to what Buzan [1993] termed "items associated with things or patterns already stored, or linked to other aspects of what is being learned" and "those items which are of particular interest to the learner". This was found to be somewhat inefficient, because many of the students did not choose to visit the information.

Overall, CAL3P was found to be popular and effective. Students were brought to a point where they could envisage the manufacture of any turned part as a series of logical steps, and attempt to write manufacturing instructions. Clearly, the software can be improved further, but the direction which such improvements should take could only be foreseen as a result of the trials conducted and the student feedback received. The most important single lesson was that skills which were imparted to students through highly interactive parts of the software such as the DCG simulation were more likely to be effectively learned.

With data on student performance collected, the development effort inherent in presenting each medium (such as video clips or animated simulations) within IEM software can be compared with their effectiveness as implied by the quiz results. It is

necessary to discover which of the techniques at the IEM software developer's disposal is the most efficient in terms of development effort required against learning outcomes achieved. This is the purpose of the next chapter. Ultimately, it should be possible to construct a model where the desired outcomes arising from the use of a multimedia programme can be matched against development requirements.

8.0 A Discussion of the Validity of IEM in Education

Chapter 6 described CAL3P, a piece of IEM software for teaching process planning. Chapter 7 detailed experiments undertaken using that software in order to measure its educational effectiveness. CAL3P demonstrated a number of possible approaches to the provision of IEM on computers; distinct sections within the software had different levels of interactivity and made use of different media such as graphics, animations, photos, text and movies.

In Chapter 7 a method was described which was used to identify students who had substantial experience of working with machine tools. With results relating to these students discounted, it was assumed that the remaining correct answers indicated where CAL3P had successfully facilitated an act of learning. The relative proportion of correct answers to each question was used to determine which features of CAL3P had taught most effectively.

It was hypothesised that the most efficient IEM methods could be identified, with student performance being measured against the costs inherent in the provision of the information, for each of the various approaches that had been tried. Using CAL3P to learn about process planning, each student had opportunities to learn facts that would later be tested. These facts might be learnt by reading text-based information on screen, by selecting specific areas as being of interest, by studying photographs, watching digital movies, or experimenting with the simulation. Each of these 'media' can be seen to have some benefit in terms of the learning outcome achieved, and each has costs, such as the development effort necessary and the technical requirements such as storage space and processor speed, which they demand of the delivery system. These are among the limitations described in Section 8.1.

8.1 Limitations to IEM Software

It was suggested in Chapter 4 that the factors limiting the exploitation of IEM could be grouped into three categories. These are hardware restrictions, cost limitations and the willingness of learners and teachers to use computer-based learning material. Each of these is considered in the subsections that follow.

8.1.1 Technical Constraints

For any software project, one obvious limitation is the capability of a typical user's computer. If the program is too demanding, the number of potential users or customers is

limited. As computer capabilities continue to increase, it might be thought that technical limitations are of declining importance, since the capabilities of computer hardware are increasing. However, Travis and Jones [1997] found that users' expectations from IEM were increasing, possibly as a result of exposure to home computer games. For example, users have come to expect high resolution screen displays with great colour depth. These require modern computers, or at least updates to video hardware, and necessitate the storage of larger graphics files. Similar hardware upgrades may be necessary if users expect to hear high quality sound, and stereo sound requires twice as much disk storage space because two audio tracks must be recorded. Users may also become frustrated with long access times caused by slow CD-Rom drives, or a slow CPU.

While the capabilities of computers now available for sale represent better value than ever before, upgrading a whole suite of computers remains a major investment. A demonstration of CAL3P at a local school revealed that the majority of their computers were several evolutionary steps behind what was regarded as the entry level for new hardware purchases at the time.

Buying a single up-to-date computer on which the multimedia authoring process is to be carried out should not be a problem, since the hardware cost is small in relation to the cost of the software development work itself. However, it is vital to consider what can be achieved on the machines typically available to learners. Features which look excellent on the development machine may be disappointing on the end user's machine. In the worst case, some users might not be able to use the software at all.

While it is good sense to have powerful hardware available to speed development processes such as applying filter effects to digitised photographs or rendering 3D graphics, it is important to test the software repeatedly on machines of lower specification. If the software is coded in a sufficiently economical way, the size of the potential 'market' for the software may increase.

CAL3P required computers of lower specification than those normally associated with modern IEM software because, as Chapter 6 described, it used program code to display 'DCG animations' which allowed CAL3P to represent the machining of an almost limitless number of different part shapes, far more efficiently than could be achieved by storing a number of different video clips. For example, the first version of the CAL3P prototype which was tested with students occupied less than 300K of disk space, yet facilitated an hour's learning. Later experience with digital video, detailed in Section 6.4.4, suggests that 300K, if used for digital video, would provide about two and a half seconds of 'instruction'. Thus, technical constraints acting upon CAL3P were comparatively minor.

8.1.2 Users' acceptance of IEM software

Despite continuing efforts to make teaching more effective or more economical through the use of IEM, inevitably there are some learners who prefer conventional methods. Section 7.5.2 detailed Cann's [1997] survey, in which learners of this kind described themselves as 'phobes'. Since a student who feels that computer-based learning is not enjoyable or not valuable is unlikely to buy software or enrol on a course which makes extensive use of CAL software, their reaction imposes a limitation on the size of the 'market' for any software which is produced.

It may be possible to win over potential users by offering a quality piece of software. Feedback from students who used CAL3P was more positive than Cann's [1997] survey (see Section 7.5.2). It is likely that the software which students use influences their disposition towards computer-based learning activities. While 'quality' in CAL software is difficult to define specifically, a program is naturally more likely to be accepted if, for example, its illustrations are attractive and informative, its exercises are logical and relevant, and the whole is free from bugs. Developing simplistic, inflexible software which features less attractive media may be a false economy.

8.1.3 Development Effort

Put simply, the creation of multimedia software is an expensive process. Each medium requires special skills for capture and presentation, and the complexity of the project increases when they are used in combination. The creation of a piece of multimedia software requires a blend of skills which are normally found in journalism, film work and other fields. From their own experiences in the development of IEM software for Engineering Education, Bhide and Nene [1997] suggested:

"Designers of a CAL package have to be a bit of an author, a visualiser, a movie director, a music composer, a screenplay writer, a special effects man, an animator and lest his package becomes boring, a clown, all rolled in one! But aren't good teachers made of all this stuff? So we relax all the above conditions and agree that it is sufficient to be a good teacher."

Bhide and Nene [1997]

While it is commendable that teachers are prepared to undertake the creation of a CAL package, the view that simply being a good teacher qualifies one for development work is not universal in the multimedia community. Vaughan [1994] suggests that a commercial IEM development project would involve a sound engineer, photographer, animator, programmer, musician, copy writer, knowledge domain expert, project

manager, file conversion specialist, ergonomics specialist and others – a view shared by Horton [1994]. It is disappointing that none of these authors include an understanding of learning theory in their lists of skills required in a multimedia authoring team. As per Hudson's [1984] warning, given in Section 3.6, it appears that much IEM development is still conducted by enthusiastic amateurs.

It is unreasonable to expect an individual or even an academic department to have the full range of authoring skills at their disposal. Modern multimedia authoring packages have reduced the complexity of the programming work required to present each medium (see Appendix 2), but the overall range of skills required (in selecting, capturing and preparing the media to be incorporated in the software) remains as wide as ever. Of course, media of a less than professional standard may be adequate for software prototyping, or for informal training in small organisations.

For a small IEM authoring project undertaken by an individual, or a high-quality project requiring the contributions of a number of people, a total number of man-hours will ultimately be committed to the enterprise. Section 4.7 introduced the problem of development effort for a piece of IEM software; if the introduction of a new IEM resource requires a massive investment of time and money, it may be impossible to justify. Again, this depends upon the size of the 'market' for the finished product.

Unfortunately, making a piece of IEM software suitable for a wider audience is much harder than creating a similar program for use in-house. For example, an IEM package may include a feature whereby details of each student's progress are forwarded to the tutor via a network connection. For use at a single site, programming to achieve this is comparatively simple, because the author will know all the details of the network which will be used. For the software to be made generic in this respect, so it can be employed on other networks, will require much more programming. It would be necessary to include a method whereby the software could be configured to suit a variety of different hardware options.

In addition to problems arising from differences in hardware (the reader may recall Odor's [1992] warning that "software rusts", presented in Section 1.3), the material within a package may need to be modified or expanded for use elsewhere. Teachers in other institutions might consider different aspects of a subject to be important, or might use different terminology. Any move to take their requirements into account will increase the length of the development process. Another problem is that software has to be of very high quality before it can be released generally. IEM software can be operated quite easily on an experimental basis, with its creators in attendance. When operated by learners at a

remote location, any problems with the software will be much more significant because they could confuse students and their supervisors alike.

As a result of the difficulties inherent in making software suitable for a wide audience, some developers concentrate on unambitious applications meant for use only in their own classes. Turnbull [1974] originally identified this tendency among teachers with an interest in CAL, to develop “My material for my kids” – an unfortunate propensity since it can result in the software which is created being used only once per year. This failure to write software which is suitable for a broader audience may be one reason for the perceived shortage of IEM titles in engineering, as identified by Bhide and Nene [1997].

The decision to develop software which addresses the needs of only a small number of users, or to target a larger ‘market’ with a more comprehensive piece of software, will depend on circumstances. In committing a larger amount of time and money to a project, a greater number of students might benefit – financial justification may be achieved in projects of varying scale.

It may be felt that issues relating to development effort will decline in importance, since new authoring languages offer faster development. Each new multimedia authoring tool claims increased productivity, though Marshall et al [1995] suggest that these promises are largely meaningless because project management in CAL software creation tends to be haphazard, making the development effort levels reported questionable. Turnbull [1974] stated that improved authoring software and more sophisticated hardware were making the development of CAL programmes easier; if multimedia authoring is becoming ever more simple, one might ask why we are still waiting for the anticipated wealth of CAL programmes a quarter of a century later.

While authoring software becomes more powerful it also increases in complexity. Offering more options to the multimedia author requires that more time is spent mastering the software and experimenting with what is possible. Thus, development effort is still a constraint on the proliferation of IEM software.

8.2 Overcoming the Limitations to IEM Development

The previous section described in detail the three types of constraint which may prevent the exploitation of IEM technology; technical limitations, users’ preferences and development effort.

CAL3P was an experimental piece of software, not in the normal ‘mould’ for IEM software. As Sections 6.3 and 6.4.5 described, CAL3P achieved animation through the use

of program code which moved entities about the screen in runtime, rather than depending entirely upon digital video. As a result, it could operate on machines of relatively low specification, avoiding the constraints typically introduced by the technical limitations of delivery hardware. This increases the number of people who might use the software.

As Section 8.1.2 described, CAL3P had proved popular with the substantial majority of students. If a piece of IEM software is perceived to be useful and enjoyable, students are likely to spend more time using it, for example by referring back to it during exam revision. A higher number of student usage hours would make the software development process easier to justify. In a commercial sense, learners might hear about a good IEM package and seek to purchase it, thus increasing the likelihood that the development project achieves payback. The other potential 'user' to whom a piece of IEM software must prove itself is the teacher. An IEM package may be perfectly functional, yet be considered unfit for use at another institution where emphasis is placed on different topics. By providing the exercise which students were required to do on paper, rather than building it into the software, teachers who might have found CAL3P useful were free to vary the level of difficulty of the exercise by changing the geometry of the part to be produced. Equally, the teacher could have allowed students to design their own part, and then write a process plan for it. Thus, the likelihood of a teacher rejecting CAL3P as inapplicable was reduced.

With problems relating to users' hardware largely mitigated, and differences in students' syllabi accommodated through the use of DCG animations, CAL3P largely overcomes two of the three types of limitation which affect IEM software. The remaining limitation to software like CAL3P is the time (and consequently money) required for software development.

8.2.1 Justifying Development Effort

Section 4.7 introduced the concept of development effort as a ratio of total time spent in the development of a piece of software against the time for which it will be used by a typical student. This is an important measure because the cost of creating an IEM package will inevitably be compared with the cost of conventional instruction. Development effort is hard to justify unless the number of users is large. Where an hour of conventional instruction might require that a teacher spend five hours in preparation, an hour of IEM instruction could require 300 hours' work, putting the conversion of even a small fraction of a syllabus beyond the reach of most tutors [Farr and Lawlor-Wright, 1997a & 1997b].

Marshall et al [1994] cite development effort ratios of 800:1 and beyond, though some estimates are far lower. Nicholls [1997] described 'ChemiCAL', a package for use with undergraduate chemistry students, which was used by 150 students for an average

of 15.7 hours each. Given Nicholls' budget of three man/years for the development process, the ratio of development to playback time is approximately 5,000:15.7, or 318:1. Since the number of students making use of the package was quite large, if it is assumed that the software remained useful for several years, the time invested in development may be tolerable. A development ratio of 318:1 appears uneconomic, but Nicholls [1997] argues that the software provides individual feedback, the computer providing instruction on a one-to-one basis. Thus, if the software has a useful life of four years it might provide six hundred students with 15.7 hours of instruction; 9,420 hours' tuition for an investment of 5,000 hours. Whether the software really is as good as one-to-one tuition is questionable. As discussed in Section 4.2, one of the disadvantages of CAL is that the computer is less flexible than a human tutor.

Nicholls' package appears to be largely text-based, and operates in a linear, predefined manner. There is no opportunity to experiment or explore alternatives. This level of simplicity can produce a relatively low development effort ratio, though of course cost-effectiveness does not guarantee that the software facilitates more effective learning than conventional instruction. Davies and Crowther [1995] warned that computer-based instruction may appear to be more efficient (with the computer performing the teaching function rather than a human tutor), while actually being much less effective.

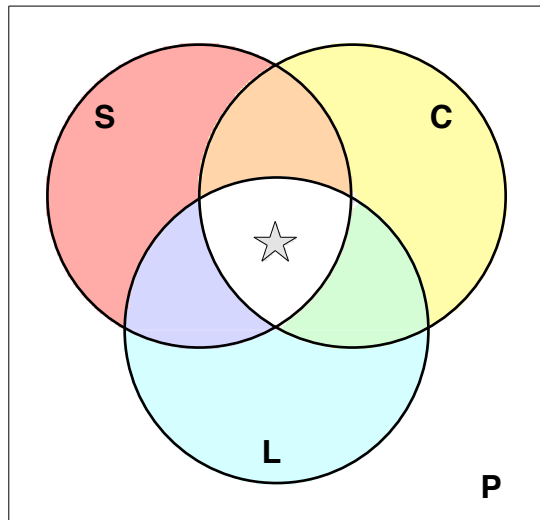
When questioned, Nicholls estimated that the development effort ratio for his software might be as low as 12:1. The calculations presented above contradict this, in accordance with the warning given by Marshall et al [1995], introduced in Section 8.1.3, that the amount of input reported in IEM development can be questionable.

8.2.2 Audience Size for an IEM Application

Given that hundreds of hours' work are required to create even the most basic IEM package, justifying the outlay involved is critical. In Section 7.0, it was suggested that an IEM package must produce better results than conventional methods, prove cheaper overall than conventional instruction, or allow learning to take place where it would otherwise be impossible. Section 6.2.1 described the shortcomings of conventional instruction in the education of novice process planners, but even if the IEM software is considered successful, it remains necessary to prove that the approach used represents a cost-effective alternative.

The only way the development of an expensive piece of IEM software can be considered worthwhile is if the number of people who will eventually use it is large. There remain, however, some issues which prevent a piece of IEM software from reaching a large 'market'.

Within the field for which a multimedia programme is intended, no piece of software will be used universally; the chosen delivery platform(s) represent a subset of the total community of computer users, and as Section 8.1.1 described, the minimum machine specification for effective use of the software will further reduce the potential audience. Learners' preferences also play a part since some people will not make use of IEM software, preferring alternative forms of instruction. Thus, the author proposes Figure 8.1 showing how a piece of software is useful to only a fraction of the population.



- P = The population as a whole
- S = All people studying a given subject
- C = All people who have access to a suitable computer
- L = All people prepared to use a computer in learning

Figure 8.1: Potential audience for a computer-aided learning programme (Only within the central area '★' is a CAL programme likely to be used)

A similar calculation could be made for any kind of software, regardless of the presence of multimedia, though set 'C' will tend to be smaller with IEM software because the programmes tend to require computers of higher specification.

Having defined the subset of the overall population who will be making use of the software, it is necessary to investigate exactly how it will be used, for the purpose of maximising the value of the instruction in relation to the constraints on the provision of instruction (contact time available, development time required, development cost, delivery platform hardware limitations).

While it is clearly desirable to make IEM software as user-friendly as possible, a second kind of 'user' must also be satisfied; this is the teacher, who will decide if the software is suitable for use by his students. For the maximum number of learners to benefit from the software, it must suit the curricula and teaching style found at other institutions. This will require increased effort during development, overcoming the "My material for my kids" bias identified by Turnbull [1974]. Since multimedia software requires much more development than conventional software, it is vital that such duplication of effort is avoided.

In 1989 the Computer Teaching Initiative (CTI) was set up. Twenty centres were formed, each responsible for a different discipline, between them covering most subjects taught at undergraduate level. Each centre worked to disseminate information, encourage the use of CAL and assist with the development of improved teaching software. Unfortunately, the report of the Working Party on the provision of computing facilities for teaching [1991] found that CAL applications were being developed within universities, but not permeating throughout the system. It was found that only five of the universities responding to a survey reported that they found CTI project software very useful, and a further thirty found it slightly useful – much of the software developed in 139 projects had not spread through the community.

To overcome the problem of software being created which solves a local training need but which is inappropriate for use elsewhere requires communication between people in each discipline, such as that facilitated by the CTI, and the desire to create software which meets the identified needs. If IEM software is to be used elsewhere it must be very stable, well documented, and comprehensive enough to suit some variation in syllabus – in each case requiring substantial development effort.

In addition to the limitations which reduce the number of students likely to make use of a package per year, it is necessary to take into account the time during which the software is likely to be useful. One major influence here is the subject matter itself. An IEM application on the subject of mediaeval history is likely to contain information useful to students for as long as homes and schools are equipped with PCs, but a package meant to train retail sales staff on the technical merits of a line of high-technology products may well be out of date in twelve months or less.

8.3 Development Effort for CAL3P

Since CAL3P's development involved only one programmer, working full-time, work undertaken was fairly simple to quantify. Naturally, no single person can have the

full range of skills associated with multimedia. Working alone in the development of CAL3P, it was necessary to acquire some appreciation of each of the developers' skills, acting as copy writer, photographer, illustrator and programmer. Fortunately, the development of both the prototype and final version of CAL3P were preceded by programming work relating to the creation of a multimedia information system for the DICTA project (as described in Chapter 5), so the authoring software employed to create CAL3P was not entirely unfamiliar. Taking time to master the authoring languages would have reduced productivity and made measurements of time spent unrepresentative.

Measurements taken during programming make it possible to apply to individual modules the same development ratio concept which is normally applied to whole pieces of courseware. To some extent, the development process can be broken down further to detail the effort required to capture and present information in individual media, this being compared to its contribution to the educational effectiveness of the software.

In the sections that follow, each of the media which were employed in CAL3P is evaluated in terms of the development effort expended in its provision, and the contribution each may have made to student performance, as measured by the built-in quiz detailed in Chapter 7.

It must be remembered that development effort is not the sole measure of the viability of a project. Indeed, some media exhibit very high development ratios specifically because they achieve highly effective instruction. For example, a carefully designed diagram might communicate a fact very quickly and in a way which learners easily memorise. It is thus an excellent teaching device, but its development ratio looks poor because a good quality diagram takes care and attention to create, yet allows the learner to absorb the information rapidly. As a result, the development ratio for the diagram compares poorly with that for a passage of text containing the same information. There can be no simple rule as to which media are most effective or economical, but an understanding of the advantages and disadvantages of each may assist the developer of IEM software in future. Also presented within each of the following sections is a compilation of guidelines on good practice when employing each medium.

8.4 Text Elements

Technically, the presentation of text is one of the simplest tasks in multimedia authoring. Most computer users are familiar with some application in which text is entered and edited, so text-based content can be generated quite rapidly. The content can

be supplied by a team of contributors, using their choice of word processor or even e-mail software.

The creation of CAL software can be quite fast if little more than the presentation of existing course notes in a computer-based format is required. At the simplest level, text can be displayed as a single page per section, with the reader scrolling up and down as required. This is a common format with which many users will already be familiar, from experience with word processors, software help files and 'Web pages.

Compared with looking at a diagram and deciding if it shows relevant information, text takes a longer time to evaluate. Being created more rapidly than other media such as images, but requiring a relatively long time to comprehend, the development effort versus 'playback time' for text appears favourable. It should be remembered that this is a result of the inefficiency of text as a communication medium, and not a desirable characteristic of the medium.

Tombaugh et al [1987] described experimental results which suggested that text shown on the computer screen commanded less attention than printed text, a view shared by Schneiderman [1987], who also found that people read text more slowly when it is presented on-line. Travis and Jones [1997] suggest this may be due to eyestrain caused by the relatively low resolution of the screen when compared with printed material, plus other problems such as screen flicker and reflections, all causing the reader to glance away from text. Other reasons for text on screen being inferior to printed text might be that an online document lacks the 'heft' of a printed document, making it hard to gauge progress, and it is seldom possible to add one's own notes to a page of text in a CAL application. These factors may cause poor performance in an IEM system which makes extensive use of text.

Text will feature in almost every multimedia application, however. It is really the only viable option for presenting headings and titles. It can also be used to supply captions, eliminating ambiguities which may be found where a single medium is employed. Vaughan [1994] suggests that written text requires less attention than narration – certainly, while reading a text-based document it is easier to pause or discuss something with a fellow learner than when listening to a recorded voice.

In one notable exception, Amore [1998] described the development of a CAL application which featured no text whatsoever, because it was found that the target users' spoken English was adequate for training purposes, but their reading skills were very poor. Avoiding text entirely was found to complicate the development process, though this was a necessary characteristic of the software, which was ultimately successful.

8.4.1 Good Practice in the Presentation of Text

Although apparently simple, text is a critical element. Words must be chosen with care, because many have more than one possible interpretation. For example, instructing a student to 'select the right answer' is potentially misleading where 'select the correct answer' is not. Text for use in IEM software requires careful writing if it is to have maximum educational effect. Horton [1994] offered a number of guidelines to the writing style which should be used for on-line documentation, of which the following are particularly relevant to IEM software:

- Use short, familiar words
- Use concrete terms (e.g. 'warning light', not 'status indicator LED')
- Avoid unnecessary computer terminology
- Don't overdo emphasis
- Avoid over-abbreviation
- Be positive (say what to do rather than what not to do)
- Use active sentences
- Be exact and consistent

Horton [1994]

Horton's guidelines are important because they serve to remind the developer that the user is not necessarily an expert, either in the subject being taught or in computing. Borenstein [1991] gives numerous examples of software in which programmers use jargon or create interfaces which, although powerful, demand a much higher level of computing experience than some users have. An IEM program's text content is largely free of technical constraints since it requires much less storage space than other media, so it is unfortunate that some software still features acronyms, abbreviations and nomenclature which is not explained.

Travis and Jones' [1997] proposed solution to the inadequacies of text on-screen was to limit the number of lines per screen, breaking down subjects as far as possible. Hudson [1984] identified a potential problem where a learner may skim through an IEM package, looking at attention-grabbing media such as photos but neglecting large bodies of on-screen text. Hudson [1984] suggests that the use 'no-response' frames (screens) must be minimised; pages of information should require some input from the user which indicates that they have read and understood the information presented.

Although the generation of basic text content for an IEM package is quite simple, following the guidelines presented in this section, to make the most effective use of the medium, requires substantial effort. A low-cost development will feature large bodies of

text, demanding only passive learning. This will be much cheaper to create, but the effectiveness of such a package is questionable.

8.4.2 Options Available in the use of Text

Until the introduction of computers with graphic user interfaces such as Macintosh (and later, Microsoft Windows on the PC), almost all applications displayed alphanumeric characters on a predefined grid of rows and columns. Early CAL software was no exception. Typically, there was only a single style for each character, later with the option of high intensity (bright) or coloured characters for emphasis.

In a computing environment which offers a graphical user interface, text is no longer so constrained. It is possible to display characters in bold or italic styles, at a variety of sizes and with a natural width that suits each letter – the text need no longer be placed on a grid. Text can be much more attractive on such systems, and representing complex formulae on screen is easier.

Modern multimedia authoring packages allow the developer to use text in a variety of styles. This increases the storage space requirement slightly because the file containing the text must also store the invisible control characters which mark the beginning and end of each differently styled section, but text still has the lowest computer storage requirements of any medium.

The use of special text styles is an issue in itself, influencing the ease with which textual information can be read and consequently absorbed. Obviously, if the developer highlights key facts, this may help the learner to find the information required – but when a programmer has little or no background in graphic design, the freedom to use a massive variety of different styles may produce unattractive results.

The choice of font may be important; decorative fonts can make a screen display look attractive, but their use for large sections of text will frustrate the reader. It was found that selecting special fonts for use, because they looked more attractive, led to problems when CAL3P was used on computers with a less comprehensive set of fonts. If a font is unavailable, Windows attempts to run the application with font substitution. Since even typefaces which are listed as being the same point size (height) can be of slightly different dimensions, font substitution was in some cases found to leave the reader with text which ran over into areas used for diagrams, or which ran off the bottom of the screen. In either case, the quality of the learning experience was impaired. It was thus important to select only from the typefaces commonly available to Windows users. Since fonts are commercial pieces of software subject to copyright issues, it is not practical to include them with IEM software. All three of the fonts which were eventually selected for use in CAL3P were a standard feature of any user's Windows installation, thereby avoiding

problems of font substitution. 'Arial Rounded MT Bold' was used on the title screen, for section headings and the user's own process plan, while 'Times New Roman' was employed for paragraphs of text such as those found in the Guide to Process Planning. For clarity, the default font 'System' was used for all buttons which had captions.

Times New Roman is a serif font; it features little decorative 'tails' at the end of each stroke which makes up the letters. Vaughan [1994] reports that serif fonts are traditionally used for body text (a page layout term referring to the bulk of the text on a page) because the serifs are said to help guide the reader's eye along the lines of text. The choice of font is not constrained by the technical limitations of the computer; using one existing font rather than another does not increase the storage space required, so the choice is simply a matter of using something with which the user will be comfortable.

Compared to layout design for printed material, displaying information on the computer screen involves a slightly different set of rules. As mentioned in Section 8.4, the resolution of the standard computer screen compares poorly to the resolution achieved with printed material. Thus, text can be hard to read, particularly the more ornate fonts when displayed at small point sizes. It was found that serif fonts were unsuitable for display below a size of about 12 points. Where text was required to be smaller, because numerous labels were required on a diagram for instance, a clear font had to be employed rather than a decorative one.

Sometimes, text displayed at large sizes looks unattractive because the spacing between letters appears uneven. This is a natural consequence of the different shape of each letter, but it becomes particularly apparent at larger sizes. Although desk-top publishing programs for the generation of printed material allow kerning (spacing between letters) to be adjusted, multimedia authoring software such as ToolBook does not. To overcome the problem of unattractive titles, it is sometimes necessary to present text in the form of a bitmapped graphic, made within an art package where the kerning could be adjusted manually. The finished result can then be saved and imported into the multimedia programme. This was particularly simple in HyperCard, which supports the creation of illustrations with a built-in painting tool.

When titles and other pieces of text are created as graphics rather than text, the storage space required is significantly higher, but the result can be more attractive. In addition to kerning adjustments, it becomes possible to use anti-aliasing, making the text look still better. Anti-aliasing is the use of transitional colours around the edge of a shape, such that its outline appears to be smoother than the actual screen resolution allows. When presenting text in the form of bitmapped graphics, it is possible to add further decorative effects, which might be desirable in a high-quality application. Of course, any

use of bitmapped graphics (discussed in Section 8.5) increases development effort, and requires more storage space.

8.4.3 Advantages and Disadvantages of Text

Naturally, multimedia involves a complementary mixture of media types, so it is not always possible to isolate a single medium as being the ideal way to convey a certain message. Text was seldom used exclusively on a page; engineering lends itself to the extensive use of images, and diagrams were used wherever they proved more efficient.

The principal advantages of text are as follows; the ease with which content may be obtained or generated, the low technical requirements for its inclusion, users' familiarity with the medium, and the ability to search for words or phrases. Cotton and Oliver [1997] found that despite the increasing importance of images, animations, sound and video, hypermedia software was still predominantly text-based. Options available to the multimedia author, even with this meanest of media, are more extensive than might be imagined, including choice of colour, size, typeface and style in order to emphasise key points or put the reader in a suitable frame of mind.

The disadvantages of text are that it does not command attention in the same way as a diagram, and comprehension takes longer. Also, the most effective use of text requires a great deal more time than is expended by simply typing material. The choice of wording and layout require considerable skill, and adding interactive elements which require the user to prove that they have read and understood the text presented (as suggested by Hudson [1984] – see Section 8.4.1) will naturally complicate the development process.

8.4.4 CAL3P's Text Content

CAL3P includes text throughout, supported by a variety of other media. There is some text on every single page, if only the section heading and button captions. Among CAL3P's ten distinct sections, in total there are 63 screens which contain what in page layout terms would be called body text (i.e., substantial sections of informative text). Table 8.1 details the use made of text within CAL3P:

CAL3P module	Number of pages	Pages with body text
Intro screen & credits	5	3
How to use CAL3P	5	5
Guide to Process Planning	10	10
Write a Process Plan	6	3
Machining Processes	19	19
Turning Principles	12	12
View Current Process Plan	1	0
View Cutting Simulation	1	0
CAL3P Quiz	11	1
Quiz Answers	11	10

Table 8.1: Text Content in CAL3P

The learner is not expected to visit every page of CAL3P, nor to carefully read every piece of text presented. For example, the guide to process planning was included as a ‘safety net’ for students who had not attended the conventional lecture which preceded CAL3P’s use. It was found that basic text was the ‘cheapest’ medium to provide, both in terms of development effort and its impact on the technical limitations of the computer. As such, text can be incorporated in quantity – provided it is well structured, such that a student need not ‘wade through’ sections while they look for key facts.

It must be accepted that much of the text provided will not be read by every student. In CAL3P the description of every machining processes included a section of text, but the student may quite reasonably visit only the pages relating to processes he wishes to use. Equally, on visiting a page, a student might glance at the supporting diagram and decide that the process is inappropriate. Thus, some text is provided which is not utilised. If the student was presented with an exercise in which he was required to write a process plan for a differently shaped part, he might read other sections of information within the software.

Purely by accident, the version of CAL3P used in the trials conducted featured a typographical error. Since the incorrectly typed word formed a valid spelling (‘with’ instead of ‘wish’), it was not detected by the authoring software’s built-in spelling checker. Two students reported the error. It is somewhat surprising how few people gave the erroneous sentence enough attention to spot the mistake, but Buzan [1974] describes how groups of words are read for meaning rather than individual substance, possibly somewhat out of sequence, by an undisciplined reader. Horton [1994] concurs, describing the reading process thus:

“The eyes do not move smoothly and regularly along a line of print. They move in a series of short jerks, with four or five glances giving enough perceptual impression to be filled out by the context. At each pause (fraction of a second), only two or three words can be perceived. Many words are skipped entirely, or not perceived clearly. Much is done by inference.”

[Horton, 1994]

Thus, it may be believed that text is not paid a great deal of attention. However, it is a vital constituent of IEM software. It does not make great demands on the computer hardware, and while it may not be read exhaustively it does allow a learner to get more information, if they consider it necessary once they have studied the other media provided. It should be noted that text is still the most appropriate medium for some information, such as lists and tables. Section 8.9.1 discusses the effectiveness of text in CAL3P, determined from student quiz results.

8.5 Graphic Elements

As Section 3.2.4 described, graphics are a particularly powerful communication method, useful for getting a learner’s attention, conveying information rapidly and offering an unequalled probability of subsequent recall.

To the human user, graphics fall into two basic categories; photographs and artificial material. Both have their uses; photographs relate the material being taught to the ‘real world’, while diagrams present simplified examples, free from the distractions which can clutter photographs.

To the developer there are also two kinds of image; bitmaps and vector graphics. Photographs are always stored as a bitmap of some kind, but in some authoring packages diagrams can be displayed in the form of a collection of lines and polygons. Under Multimedia ToolBook, CAL3P and the DIS both made extensive use of vector graphics, which allowed large, clear diagrams to be presented with minimal storage space. Diagrams can also be stored as bitmaps if necessary, such as where the diagram is very complex or where it has been scanned from a paper-based original.

Unlike the creation of text content, the generation of graphics for IEM software requires skills which are not common to every multimedia author. Many authoring packages have some simple, built-in drawing utility which can be used to create basic diagrams, but these take time to master and still require some artistic ability if images are to achieve their full potential. More detailed images require additional software and specialist skills such as photography.

8.5.1 Good Practice in the Presentation of Graphics

Images should be displayed with other media. An image may catch a learner's eye, but labels, annotations and comments will then help a user understand the image. Commentary, either text or audio, must correspond with the images which are shown.

As with text, the limitations of the computer display impose some rules not normally encountered by the graphic designer. Diagonals should generally be avoided, because the computer screen produces what Burger [1993] refers to as 'jaggies' or 'the staircase effect', where lines which are close to the horizontal or vertical can seem to have distinct steps which look quite unattractive. (For bitmapped graphics with a high colour depth, anti-aliasing can be employed to mitigate this effect.)

Naturally, since development cost and storage space are constraints, purely decorative images should be excluded from IEM software. It is possible to have too many graphics, and, as with any medium, graphics may inhibit learning if used in a particularly distracting way. Vernon [1971] warns that "unusually vivid and emotional incidents may distort the assimilation of the content of the programme as a whole."

8.5.2 Options Available in the use of Graphics

One of the basic choices to be made when designing the screen layout for a piece of IEM software concerns the screen resolution. Resolution is measured in terms of the number of pixels (picture elements) which the screen shows, in width and height. Modern desktop computers are capable of displaying a variety of resolutions, typically up to 1024 x 768.

CAL3P and the DIS are meant to operate at a resolution of 640 x 480, a common display size which almost any PC now in use can achieve. When a monitor is set to display at higher resolution, CAL3P appears as a window of less than the full size of the screen. If the window is stretched or maximised, CAL3P is displayed with a grey border around the edges of the 640 x 480 area used. If CAL3P had been designed to feature page sizes greater than 640 x 480, low-resolution users would have seen a partial screen display with scroll bars for navigation – making it much harder for a learner to read the information presented, and click on buttons. Since CAL3P's other technical requirements were low, it was judged unnecessary to alienate users with computers which could not support high resolutions.

Colour is an important issue for many illustrations on the computer. The level of colour supported by computer hardware is specified as the number of different colours the computer can display, based on the number of bits used per pixel. Various different file formats may be used to store images, and with many authoring packages the developer

has a choice of several, allowing storage space to be saved at the cost of some image quality. Section 8.5.3 describes the impact of colour depth on storage space required in more detail. CAL3P employs 24-bit colour, and thus looks best when viewed on a computer which can display such a range of colours, though it can be used on less advanced machines, simply looking less attractive.

8.5.3 Advantages and Disadvantages of Graphics

As Section 3.2.4 described, images are a very powerful communication medium. Vernon [1971] concurs; "...man makes so much use of shapes drawn on flat surfaces that his ability to comprehend these has reached a high degree of efficiency." Clearly, graphics are one of the best ways to convey information to humans, but computer hardware is not ideally suited to the presentation of graphics, due to small screen size (relative to the angle of normal human vision), the relatively low resolutions which are common and the amount of storage space required for digital images. Also, special skills are required to create images for use in IEM software.

Storage space for graphics can be substantially greater than that required for text. Figure 8.2 shows a simple piece of information, conveyed in two different ways. The graphical version demonstrates the advantages normally associated with images; it attracts the attention of a reader who is skimming through a document and is likely to be remembered. For readers who don't speak English, the pictorial representation should still have meaning. The graphic is also less ambiguous than the text; the image is suggestive of an old steam train, while the single word 'train' supplies no additional information. In fact, it might be interpreted as a verb, or in some other way. Thus, it can be seen that in some circumstances graphics are more effective than words. Nobody would suggest producing a maintenance manual with no illustrations, nor attempt to convey the contents of an Ordnance Survey map with descriptive text alone. Images have a valid place in both paper-based and computer-based documentation.

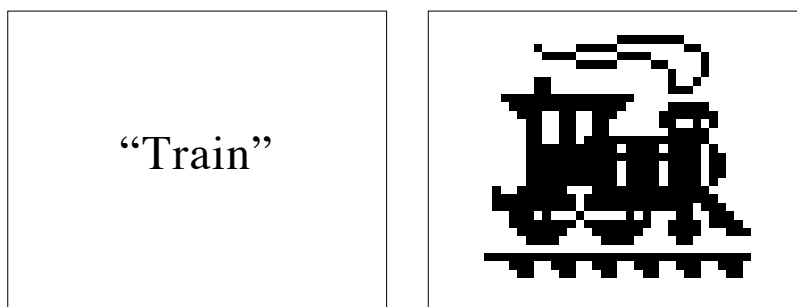


Figure 8.2: Two different ways of representing a simple piece of information

Against this, if graphics are to be employed, they must be created or obtained. Clip art libraries will not always have a suitable picture, and most other published images are protected by copyright law. An inappropriate image might cause confusion, but the production of something more suitable could be time-consuming and expensive.

One major pitfall is the relatively large amount of computer storage required for images. The simple black and white icon of the railway train presented in Figure 8.2 is a very poor quality image given the capabilities of a modern computer. The icon was drawn on a grid of 32 x 32 squares, each of which could either be wholly black or wholly white. Thus, it contains 1,024 separate elements and requires 1,024 bits of storage space, or 128 bytes. The text 'Train' required just five bytes. Even though text may also require a few control characters for formatting, as Section 8.4.2 describes, the 128 bytes used for the graphic representation offer enough storage space for perhaps three sentences of text. From the computing requirements viewpoint, text is clearly a more efficient way to store information, particularly since most graphics employed in modern IEM will be larger, and will have greater colour depth.

The train icon in Figure 8.2 is 32 by 32 pixels and used a colour depth of 1 bit; each pixel could be only black or white. With a colour depth of 8 bits, 256 hues or levels of grey can be represented in each pixel. This level of colour depth is common for icons in modern multimedia authoring software; photos may require a greater degree of fidelity. In 8-bit colour, an icon the size of the one shown in Figure 8.2 would require 1,024 bytes (1 kilobyte) of storage space.

For high quality images 24 bits per pixel can be used, with three groups of eight bits representing the levels of red, green and blue light at each pixel. These can be combined to form hues which are sufficiently subtle as to be considered 'true' colour representations for most normal applications. Unfortunately, this image quality comes at the cost of increased storage requirements. To use 24 bit colour depth on a screen of 800 x 600 pixels (a common modern resolution), a total of 11,520,000 bits will be required (i.e. 1,406K or 1.37 Mb). Thus, one image of screen size would fill a 3.5" high density floppy disk.

Image compression techniques seek to economise on storage requirements by encryption, identifying areas of an image where the colour is constant and thereby storing the data more efficiently. Heath [1996] describes a number of techniques which can be employed to reduce the size of a file, some of which degrade the quality to a certain degree, while others encode the data more efficiently but require additional time to unpack. Vaughan [1994] terms these as lossy and lossless compression schemes respectively. Since the human eye (or ear, for audio files) may not miss the data which is

discarded, a slight degradation of media quality can be tolerated. The 1.37 Mb image described above, compressed under the common JPEG (Joint Photographic Expert Group) format, might occupy around 400 kilobytes of storage space, which is probably tolerable for multimedia software supplied on CD-Rom.

This overview of computer storage methodology for graphics illustrates the significant differences between the human and the computer. While computer memory can be used most effectively when information is abstracted into strings of text and numbers, the human brain is better able to recall information which is presented in the form of images. The developer of an IEM must decide if the storage space and development time available allows extensive use to be made of graphics.

8.5.4 CAL3P's Graphic Content

CAL3P includes a great many graphics, even discounting the simulation module which can theoretically produce an unlimited variety of screen displays. To assess the graphical content of CAL3P, it must first be broken down into bitmaps and vector graphics, as described in Section 8.5.

Bitmaps were employed for the display of photographs, of which there are eight within the software. Apart from a small 'University of Salford' logo, all other images were presented as vector graphics, reducing the storage space required for CAL3P and increasing the speed of the software. It was found that when moving to a new page of information, those featuring full-colour photographs took a long time to appear on slower computers. To overcome this problem, most photographs were offered on a 'pop up' basis, where the learner was invited to click on a 'camera' button to display the photo. This way, if a user visited a page repeatedly, they did not have to wait each time while the photograph was loaded from disk and displayed.

Because the vector graphics were created specifically for use in CAL3P, each was able to show, without unnecessary complication, exactly what it was felt the student needed to know. A real lathe is a very complex piece of equipment, with levers and handles which allow numerous different machining operations to be carried out, but CAL3P is about process planning, not lathe operation as such.

It was undesirable to distract the student by presenting him with a very complex model of a lathe, not least because any machine tool he might operate subsequently will inevitably have different features. Figure 8.3 shows a photograph of a lathe, and Figure 8.4 shows one of CAL3P's generic representations.

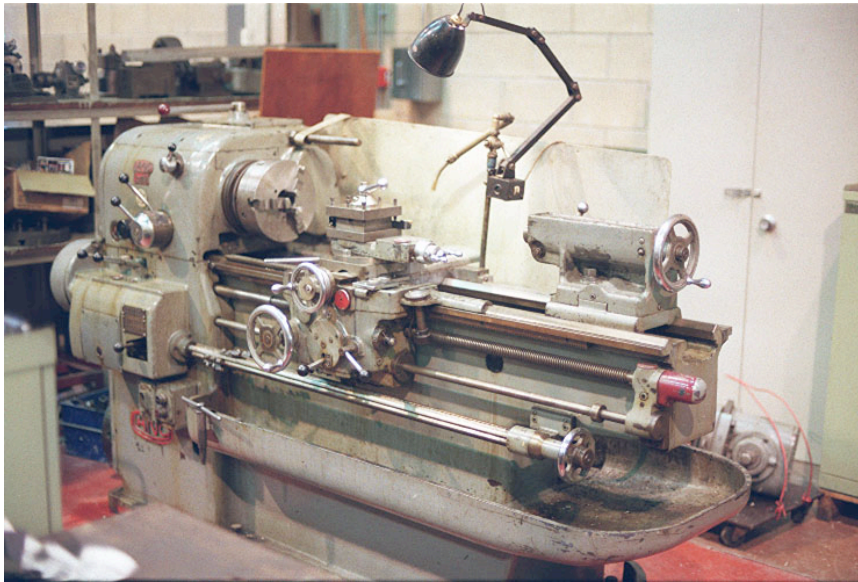


Figure 8.3: Photograph of a Lathe

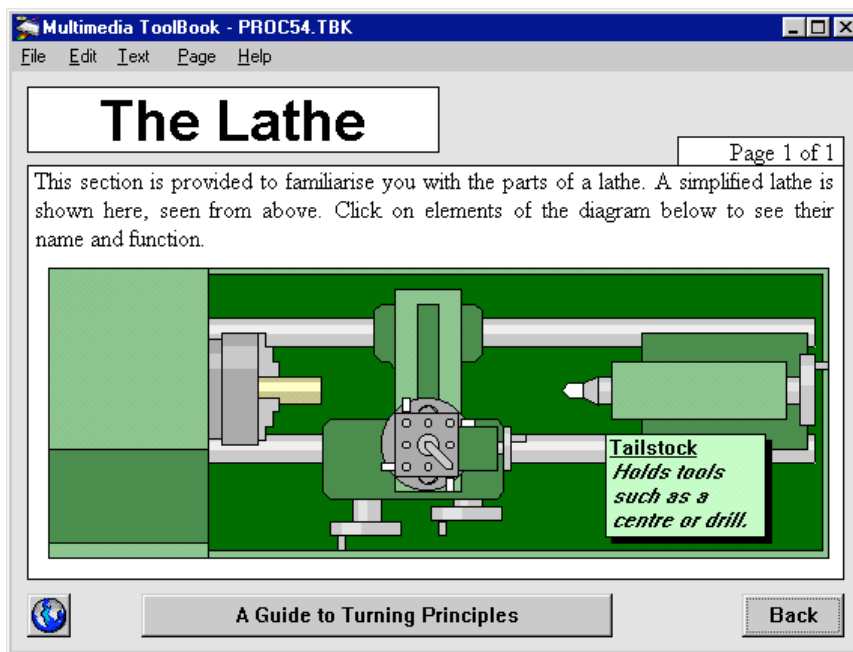


Figure 8.4: Screen display incorporating a vector graphic of a generic lathe

It might be imagined that taking a photograph is quicker than drawing a diagram, but this was not always found to be the case. Digitising and subtly modifying a photograph to make it suitable for inclusion in IEM software can take hours. For a photograph to be most effective, it must be edited in a powerful graphics package which allows contrast and colour balance to be adjusted to make the most of the image; most of the pictures taken for use in CAL3P acquired a strange colour cast because the lighting in

the laboratory was poor. (Photographs taken under strip lights look green, and those taken under ordinary lightbulbs acquire a yellow tint.) Where a flashgun was used, stark bright areas and hard shadows appeared, sometimes obscuring key features. Finally, some photographs featured unwanted details which might have distracted the learner. All these problems could be corrected using image editing software such as Adobe Photoshop, although this process requires time and experience. Figure 8.5 shows how a photograph of a lathe was improved with about half an hour's work. The original picture looked quite 'flat', with the shelving in the background appearing to be part of the lathe itself. Changes included cropping the image so that the lathe itself occupied a greater part of the picture area, selectively blurring and darkening unwanted details and increasing contrast slightly where detail was important.

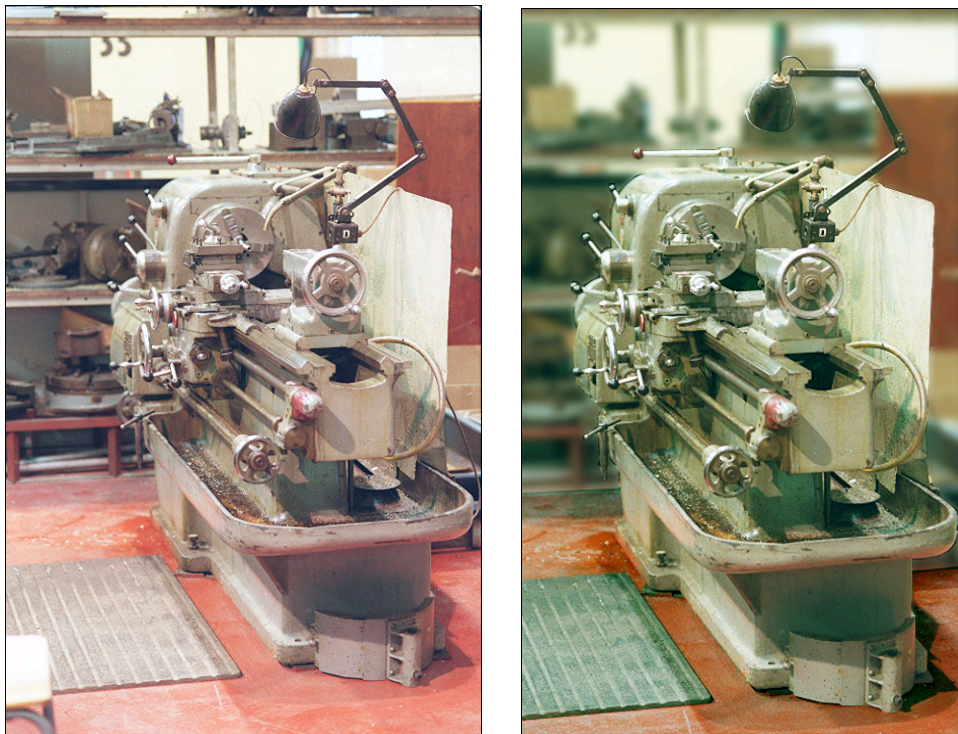


Figure 8.5: Original, and modified photographs of a lathe

Table 8.2 shows where images were employed in CAL3P. The module 'View Cutting Simulation' does make use of vector graphics, but these formed part of an animation to reflect the user's process plan entries, and are thus addressed in Section 8.7.

CAL3P module	Number of pages	Pages with photographs	Pages with vector art
Intro screen & credits	5	1	0
How to use CAL3P	5	1	3
Guide to Process Planning	10	0	6
Write a Process Plan	6	0	4
Machining Processes	19	5	19
Turning Principles	12	0	9
View Current Process Plan	1	0	0
View Cutting Simulation	1	0	0
CAL3P Quiz	11	1	8
Quiz Answers	11	1	7

Table 8.2: Use of Images in CAL3P

There are a total of eight photographs in CAL3P (the quiz and answers section both use the same photo), and 57 pages which display vector graphics. The large number of images was deliberate, making use of pictures wherever they could supplement or replace descriptive text, following what had been learnt about the suitability of images to the human learning process.

Experience gained during development suggests that including a good quality photograph in an IEM package may require two or three man hours of development effort. Once developed, many photographs were found to be less suitable than had been imagined, resulting in substantial image editing work, or further photography. The emerging digital camera technology may reduce this, once equipment of suitable resolution and colour depth falls within the budget of the multimedia developer.

When creating diagrams, it was often found possible to copy an existing diagram and adapt it for use on a new page. In this way, development time was minimised and illustrations acquired a distinctive style. Modifying vector graphics was much faster than performing a similar operation on a bitmapped graphic. Making use of existing graphics where possible, a typical vector graphic for CAL3P could be created in about an hour.

Vector graphics also formed the basis of CAL3P's animated machining simulation, as described in Section 6.4.5. In this chapter, animation is considered as a separate medium, discussed in Section 8.7.

8.6 Video Elements

Digital video presented on the desktop computer is perceived by some users and developers as a key characteristic of computing if it is to be labelled as 'multimedia'.

Heath [1996], for example, describes digital video as "... a single key operation that any multimedia system has to perform which makes it different from a normal PC." Certainly, video can be more compelling than other media; Vaughan [1994] describes how a movie of John F. Kennedy's "Ich bin ein Berliner" speech would teach more effectively than a scrolling text field containing the same words.

Video is the most technically demanding of the constituents of multimedia. Section 8.5.3 described how a full-screen still image might occupy up to 1.37 Mb of storage space; for television quality moving pictures 25 frames per second will be required, giving a total requirement of over 34 megabytes per second – and a processor fast enough to manipulate data at that rate. As with still images, compression methods are available which reduce the storage space required but degrade the picture quality to some degree.

As the most demanding medium, video was the most recent addition to the arsenal of the IEM developer, which may account for the notion that the inclusion of video is vital in multimedia. However, as the subsections which follow show, digital video must often be of limited quality, and can be surprisingly hard to use to maximum effect.

8.6.1 Good Practice in the Presentation of Video

Shooting good quality film clips is a skill in itself, which the members of the development team may not possess. Holding the camera steady, framing the subject well, focussing and making effective use of lighting are all important techniques – particularly since they influence the quality of the final product without requiring subsequent development effort or additional storage space.

Even at its simplest, filming video for use in multimedia is slightly different to filming home movies; since playback resolution will be low when video is to be displayed in a small window, camera work should concentrate on close-ups rather than panoramas.

As the next section describes, choices must be made which limit the quality of the video, but which make it easier to view. Full-screen video at 25 frames per second could look excellent (digital video of that specification would be of comparable quality to broadcast television), but few users would own the hardware necessary play back such data-intensive files. Most digital video clips occupy only a fraction of the screen area, and use a lower frame rate.

8.6.2 Options Available in the use of Video

Since video is such a technically demanding medium, most of the options available involve limiting the quality of the film clip in order to save disk space. Firstly, the resolution at which the digital clip will be stored must be chosen. The resolution of a video clip is measured in pixels, like screen resolution. Some applications, including

Multimedia ToolBook allow the 'stage' (video window) to be stretched to a different size, so a 160 x 120 resolution clip might be played in an area 240 x 180. A movie presented thus looks grainier, but fills more of the screen; actually digitising the video at a resolution of 240 x 180 would require more than twice as much storage space.

Many compression formats are available for use with digital video, reducing the storage space required and structuring the data to allow smooth playback. Some allow the developer to nominate the degree of compression; Vaughan [1994] suggests that compression of 20:1 or even 50:1 can be achieved without visible degradation of image quality.

One option which should not be overlooked is that a digital video file need not come from an analogue source such as a video tape; material ultimately presented as digital video may originate as computer artwork or rendered 3D graphics produced in a modelling package. These have the advantage that the creator can depict exactly what is required, without having to find the right subject in real life – though 3D modelling may involve substantial development effort.

8.6.3 Advantages and Disadvantages of Video

Terman and Merrill [1937] suggested that film displayed vitality and immediacy, getting attention and possibly showing things the viewer would not be able to experience otherwise – outcomes which it is now hoped to achieve through the use of IEM. Digital video can demonstrate all the advantages of conventionally presented educational films, with the additional benefit that in a computer-based format, watching a film can easily be interleaved with other learning activities. Terman and Merrill [1937] warned that there was no guarantee a pupil watching a film clip understood what he saw. It was recommended that commentary should also be used, and that presentation of an instructional film should be structured to include an introduction, and a discussion afterwards. IEM software can present training which follows these principles.

Requiring extremely large amounts of storage space and a fast processor speed, video may limit an IEM package to use on only the most modern computers. With this medium, even the substantial storage space of a CD-Rom can be filled quite easily, so IEM software which makes extensive use of digital video may well have to be largely linear. Reduced interactivity may reduce the educational effectiveness of the software as a whole.

8.6.4 CAL3P's Video Content

CAL3P incorporated a number of short video clips, showing machining processes. Although many processes were represented by animation (discussed in Section 8.7), the film clips were important because they showed a real machine tool, allowing the learner

to relate the clean, simplified graphics found in CAL3P's diagrams and animations to reality. As Section 8.7 describes, animations had their advantages, such as flexibility and the low amount of computer storage space required, but they were unable to convey effectively the violence of real machining; how coolant and swarf fly, and the noises which result. Digital video clips were investigated as a potential solution. At various places within the CAL3P software, a button with an icon of a movie camera was shown. Clicking on these showed a short film of a machining process. Section 6.4.4 described the video clips which were used in detail; Table 6.1 lists the duration and file size for each.

Development effort inherent in the presentation of CAL3P's six AVI movies was a total of approximately twelve man-hours (a technician was also employed for some of this work), and playback time for the movies, assuming a student views each one once, is 113.1 seconds. This represents a development time to playback time ratio of 382:1, not including time to code 'show movie' buttons into various screens of CAL3P. Chapter 1 suggested that development effort may be a major impediment to the successful exploitation of multimedia technology, and experience with CAL3P's digitised movies does nothing to discredit this statement.

It was found that movie files produced by rendering in a 3D modelling package required much greater input than simply pointing a video camera at an existing artifact, and this technique had problems of its own, as described in this section. It is estimated that five hours were spent constructing the model shown earlier, in Figure 6.21, and defining movement paths for the animation, representing a development effort ratio of about 1800:1. Theoretically, subsequent animations should be a little easier to produce because some parts of the existing 3D model might be reused. Having defined an animation, the computer had to be left alone for several hours to perform the rendering work. Since this process does not require human input, it does not constitute development effort as such, but it still represents a bottleneck. Once rendered, animations often proved to be unsatisfactory, requiring further experimentation with lighting, textures and video file compression options in the 3D modeller. Any changes would be followed by another rendering run. Overall, development effort for 3D models was found to be greater, and resulted in less informative video sequences than were achieved by digitising videos of real machining operations. (Of course, for some subject matter, filming the real thing might be impossible.) The effectiveness of CAL3P's video clips, as determined from student quiz results, is discussed in Section 8.9.3.

8.7 Animation Elements

An animation in this context means a screen display which features motion, but which is not achieved by showing a movie clip. It does not refer to digital movie clips which happen to feature artificial content, such as cartoons or rendered movies. The format in which the animation is stored supplies a useful distinction here; movie files (such as those in .mov, .avi or animated .gif formats) can be considered movies, not an animation, despite the fact that its content may be artificial rather than photo-realistic.

Section 8.6.3 described the disadvantages of using video clips, most notably the large file sizes involved and the computer speed demanded for even the poorest quality movies. This section discusses scripted animation as an alternative method of showing how an object moves or changes over time.

8.7.1 Good Practice in the Presentation of Animation

Producing an attractive animation basically requires the same skills as the design of static artwork, such as the use of complementary colours. Anybody who can produce good still artwork could team up with a programmer to produce informative animations.

With regard to good practice in programming, the user should not be forced to view an animation; they might return to a page to check a single fact in the text, and will be displeased if they have to wait while a long animation runs its course. Animations should be programmed so that the learner is able to break out of an animation at any time. Alternatively, the animation could be linked to a button so that it only starts when the user wants to see it. The former is preferable since it makes debugging during development somewhat easier.

As with any medium, animation should be relevant if it is to be used at all, since otherwise it might distract the learner, while also requiring substantial development time.

8.7.2 Options Available in the use of Animation

If animation, as defined in Section 8.7, is to be employed then the first choice has already been made; to achieve the display of motion without the use of a video clip. Having decided to use animation, the degree of freedom available will depend on the authoring software which is employed.

Two common kinds of animation can be distinguished; scripted animation and path animation. Path animation is a good choice for novice or non-programmers; the motion of an object or group of objects follows a trajectory defined by the developer as a number of points, typically by clicking at each point in sequence with a mouse. The smoothness of the 'curve' that the objects appear to follow depends on the patience of the developer.

Path animation is the only kind of animation which can be achieved in multimedia authoring software which does not allow coding. Path animation was employed for a simple display within the DIS software, but the result was of poor quality in relation to the time invested. Thereafter, animations were achieved by coding, as an example presented in Section 5.7.1 showed.

Scripted animation involves the use of program code to change the properties of objects. The possibilities depend on the authoring software being used, but Multimedia ToolBook is a good example, allowing the position, size, shape and colour of objects to be altered. A massive variety of animations can be displayed in this way, though complex animations require substantial development effort.

Other kinds of animation which might be employed include palette animation and font animation. In palette animation, some of the set of colours which an application uses are redefined rapidly and repeatedly to make elements within the screen display appear to flash or move. This allows a carefully drawn still image to give an illusion of motion. For instance, a picture of a waterfall might use a palette of sixteen colours, including several different shades of blue. When the RGB values of the shades of blue are cycled, the water can be made to appear to flow. This is an excellent technique for achieving the display of moving pictures on old equipment, though it requires a substantial amount of time and artistic skill.

Font animation makes use of the computer's power to manipulate text rapidly. Vaughan [1994] describes how font editing software might be used to store a number of images as if they were characters within a new typeface. By repeatedly changing the 'letters' displayed in a text field which makes use of the special font, an image is given the illusion of motion. Limitations on this technique are that special font editing software must be obtained and mastered, that the maximum size for a letter is fairly small, and that all fonts are presented in monochrome.

8.7.3 Advantages and Disadvantages of Animation

As diagrams sometimes present a useful alternative to photographs, so can animation replace movie clips. Where a photo or movie might include unwanted features which could distract the learner, an animation can be programmed to include only the details which are required. Horton [1994] suggested that employing diagrams rather than photographs can make small, crucial items easier to recognise, and it seems reasonable that this principle could also be applied to moving images. When employing animation in place of digitised video movies, the IEM developer also has greater freedom to select exactly what they want to show within the software, with no need to go and film the thing 'for real', which in some cases may be expensive or impossible.

One disadvantage of animations is that they are somewhat artificial, and may be over-simplified. For example, it is difficult to represent people adequately, yet to leave them out entirely would be unnatural in many situations. Where animation would be very difficult to achieve, video clips may be a better solution.

Animations like those in CAL3P require specific programming, and hence some programming skill. This is not necessarily within the capabilities of an IEM developer; several modern multimedia authoring packages are marketed on the basis that they allow development without actual programming (Appendix 2 details authoring software). For authoring software which does not allow scripting, or for a developer using a package like Multimedia ToolBook but only making use of predefined 'wizards' to avoid coding, only path animation will be possible.

This is unfortunate since animation is a very flexible medium which can produce vivid, memorable images for a very small amount of storage space. For example, in an IEM application teaching applied mathematics, it might be desired to show a rubber ball bouncing up and down, supporting a section of theory. One way to present this would be to set up a camcorder and film a ball being dropped. The video would then be converted into a digital movie file of some suitable size and resolution, and included in the software. A second option would be to draw a ball (or use a piece of clip art, or digitise a small part of a photograph) and have this icon moved, either by predefined path animation or by using program code.

Experience with CAL3P's digital video sequences showed that a typical small movie clip might require 123K per second, even though it only filled a small fraction of the screen. If scripted animation were employed instead, the amount of storage space required could be much lower. A non-programmer using Multimedia ToolBook could make the ball bounce using the path animation utility, carefully clicking in sequence at each point where he wished the ball to appear. Within ToolBook, this process produces a script (though the 'programmer' need never see it) like the one shown below:

```
to handle buttonclick
  show button "ball"
  move button "ball" to 1000,1000
  move button "ball" to 1020,1050
  move button "ball" to 1034,1110
  move button "ball" to 1043,1180
  move button "ball" to 1048,1270
  ...
  ...
  move button "ball" to 2500,4300
```

```
hide button "ball"  
end buttonclick
```

Clearly, to have the ball bounce smoothly requires many entries, and the script could become very long indeed. Even so, a path animation would occupy much less disk space than a digital video clip of the same duration, requiring perhaps 1K per second. A more experienced programmer using Multimedia ToolBook to animate the bouncing ball might produce a script like the one shown below:

```
to handle buttonclick  
  show button "ball"  
  i = 1  
  while i < 70  
    position of button "ball" = 1000 + (i*65), 4000 - (abs(sin(i)  
/ 0.0175 *  
200 / (i^1.05))  
    i=i+1  
  end while  
  hide button "ball"  
end buttonclick
```

The result is a very smooth animation which occupies minimal memory and which, unlike digital video, does not require a fast computer processor. (In fact, it might be necessary to include a time-based pause command, to slow down the script's execution on the fastest computers.) In allowing slow computers to display moving displays, animations are superior to digital video because a slow computer displaying a movie would have to skip frames in an effort to maintain synchronisation with the soundtrack, a technique which can severely reduce the effectiveness of a movie clip.

Still further improvement is possible; the animation script presented above could be replaced by a simple simulation, in which the user is invited to specify the mass of the ball, a value for the pull of gravity, the elasticity of the ball and its initial velocity. This would require more development effort than the example presented above, but the greater degree of involvement would make the lesson much more memorable to the learner, and it would still tax the computer far less than the display of a digital video.

8.7.4 CAL3P's Animation Content

Although the DIS employed both scripted and path animations, with several of the former, CAL3P offers only one animation, in the 'View cutting simulation' module.

However, this one animation is very complex since it draws on process plan entries by the user and displays the outcome dynamically, as described in Section 6.4.5.

The main script for achieving DCG animation is some 750 lines long, and is supported by several related scripts and a number of hidden fields which store variables. It was found that displaying DCG animations required the greatest development effort of any part of CAL3P. Seventeen different turning operations are supported, each of which may be given various parameters by the user. For example, an apparently simple process like closing the chuck jaws around the bar requires the control of a number of variables, and a number of tests must be made. Scripted checks which are made to ensure that a student's instructions are viable include:

- Rejecting a grip instruction when something is already in the chuck
- Rejecting an extension which is greater than the length of the material
- Warning when a grip instruction does not place at least 80% of the jaws' surface area in contact with the bar

Still more checks should be done, preventing such errors as trying to grip a tapered surface, ruining a high quality surface finish by clamping the jaws of the chuck onto it, or perhaps clamping a very thin-walled section of material which could deform. Naturally, all these features require time to implement and test, further increasing the development effort inherent in the software's creation.

As Sections 4.7 and 8.2.1 described, development effort can be measured in the form of a ratio of development time versus playback time. This was examined with respect to DCG animation, as with each of the media which CAL3P employed.

Gripping the workpiece is one of the simplest pieces of DCG animation, but it serves to illustrate the amount of development effort inherent in the provision of IEM software. It is estimated that a total of five hours was spent on the development of scripts relating to the 'grip in chuck' operation. The total playback time on-screen while a student sees their operation 'Grip in chuck, extension = 88' carried out is about five seconds. Comparing development time to playback time, DCG animations appear to have a development effort ratio of 3600:1! This suggests that the effort inherent in providing this form of educational media is entirely prohibitive, given Marshall's [1994] finding that development effort was typically in the region of 50:1 to 800:1.

It must be remembered, however, that CAL3P's animations are designed to be viewed many times. A conventional piece of IEM software presents media which are identical every time they are displayed, so the software is unlikely to be used repeatedly.

Process planning with CAL3P requires an iterative approach; entering a process plan, watching a simulation of the plan being carried out and then modifying the instructions as necessary. A student might well run the machining simulation a dozen times before an exercise is satisfactorily completed. Thus, he sees the 'grip' animation twelve times, varying in response to changes in his process plan. The ratio of development effort to playback time is now much healthier, at $(5 \times 60 \times 60) : (12 \times 5)$, or 300:1. If CAL3P were to be used again, to write a process plan for a different part, the development ratio would become increasingly favourable.

The effectiveness of CAL3P's animated machining simulation, as inferred from student quiz results, is discussed in Section 8.9.4.

8.8 Audio Elements

The capability of reproducing sound formed an important part of the technical standards which were set up to define what constitutes a multimedia computer (such as MPC and MPC2). Sound is an important medium through which most people receive a great deal of information, by talking face-to-face, using a telephone or listening to the radio, etc. Spoken language and music predate writing, so perhaps in this medium we have also attained the "high degree of efficiency" which Vernon [1971] ascribed to our use of images (see Section 8.5.3).

In a multimedia programme, sounds might be incorporated in the form of music, speech or other recorded sound effects. A statistic from the British Audio Visual Society which has been used in the past to advocate multimedia software suggests that we remember roughly 10% of what we read, 20% of what we hear, 30% of what we see, and 50% of what we see and hear [Leonard, 1994]. Sound, it is suggested, is more effective than text, and the presence of a soundtrack boosts the effectiveness of visual media. As section 8.6.3 described, Terman and Merrill [1937] contended that the meaning of pictures was inferred, but commentary was directly stated and focussed learners' attention. As such, sound is a potentially powerful medium, though as with each of the media available to the multimedia developer it is subject to limitations. Perhaps the most obvious limitation for sound is that it is a time-based medium; a large diagram can be used by looking selectively at individual parts which are of interest, but it is much harder to locate relevant parts of a long speech or sound recording. The subsections which follow describe how best to make use of sound within IEM software.

8.8.1 Good Practice in the Presentation of Sound

When incorporating sound in IEM software, it is important that the user still has control. He might visit a page of information only briefly, to check a single fact, or while en-route to another hyperlinked page. In such a case, the user will not want to be kept waiting while a lengthy sound file runs its course.

It may be best to leave the learner to choose when sounds are played. The developer can provide a 'play' button or even a group of control buttons including functions such as 'pause' and 'rewind'. (A standard set of symbols for these functions has evolved with which most users will be familiar.) When a page of information is left, any sound file underway should stop playing, since it may not be relevant to the new page. This will require specific programming.

Although final audio fidelity comes at the expense of computer storage space, as Section 8.8.2 describes, audio quality can also be influenced at the recording stage, by using superior recording equipment, eliminating background noise, employing a skilled sound technician and selecting appropriate material. When an IEM program features speech, it must be carefully scripted and should make use of an experienced actor. Too often, narration is reluctantly supplied by a member of the development team whose oration is inferior to that which learners normally encounter via television and radio.

8.8.2 Options Available in the use of Sound

To the developer, once audio content has been selected and captured, the level of quality at which it will be digitised and thus played back must be selected. High quality sound can take up a lot of storage space. 'CD quality' (actually ISO 10149) is a commonly recognised benchmark for high quality digital audio. It achieves stereo by storing two tracks, each sampled with 16-bit resolution at a rate of 44.1KHz. The total storage requirement is 10.5Mb per minute. A CD-Rom can typically hold 650Mb of data, but for IEM software this space is also required for images, text, movies and other files. Thus, the developer may seek to record sound at a lower quality in order to save space.

The sound sampling rate of 44.1KHz might be halved; the resolution might be reduced to 8-bit, or the sound might be recorded in mono – any of these changes may go unnoticed if the playback equipment is of poor quality. Certainly, the basic message can still be supplied, even at sampling rates of 11KHz.

To present music on the multimedia computer, as an alternative to playing digitised sound waves, the MIDI (Musical Instrument Digital Interface) format might be used. MIDI allows music to be stored in an encoded form, where each instruction in a file corresponds to an action, such as the press of a piano key. The advantages of this are that files are much smaller than digital audio, can easily be edited (with the right software and

skills), and do not place great demands on the computer's CPU when played. However, the MIDI format can only store and reproduce notes from a given set of instruments, and cannot support spoken dialog (or singing). Also, different playback devices may interpret MIDI instructions slightly differently, so the music might not sound quite the same on the machine which is used for playback.

One option which was briefly explored in the DIS was to achieve narration using text-to-speech software. This was investigated because it was thought it might offer a way to obtain the benefits of narration (as described in Section 8.8) without the sizable amounts of storage space needed for long audio files. A ToolBook script was written which sent instructions to a text editor called Textó'LE, using the Windows Object Linking and Embedding (OLE) protocol. Textó'LE can read text-based information aloud using a synthesised voice. It was aimed to have Textó'LE run in the background while the user saw only the DIS, and this was achieved. It soon became clear, however, that the synthesised voices available within Textó'LE were a distraction, because they sounded strangely distorted. Also, Textó'LE did not always pronounce words perfectly. Writing separate text strings with phonetic spellings would solve this problem, though in that case the narration would no longer be acquired 'for free', since writing the additional phonetic version of the text would require additional development time. Another problem was that linking with external applications often caused the ToolBook software to freeze for unknown reasons. This could only increase in frequency if the DIS were distributed for use on computers which were configured differently, so it was decided that further development work should be directed towards self-contained IEM software. For now, it seems that narration is best achieved with a digital sound file. It is likely that synthesised speech software will be improved in the future, so its integration with IEM software may be worth further investigation in the future.

8.8.3 Advantages and Disadvantages of Sound

A clear advantage which audio has over text is that it does not require that the user studies it closely. A user who is looking away from the screen (or looking at another part of the screen) can still receive information through narration. Against this, audio is quite rigidly linear. A reader can skim through sections of text looking for something relevant, but cannot do the same with a narration.

Travis and Jones [1997] assert that making use of the sound capability of multimedia computers is not entirely practical in a classroom situation; machines equipped with speakers cause distraction to other students in the computer lab. Even when each computer is used for the same application, they will not be in synchronisation, so cacophony can still result. Headphones may be thought to present an alternative, but they

are fragile, at risk of theft and a potential hygiene problem. Also, few computers have sockets for more than a single pair of headphones, posing a problem for group work. These limitations do not apply where the IEM software is likely to be used in a distance learning context. Here, only a single computer is likely to be found, so it is unlikely that any audio which the computer supplies will distract another learner. Sound (and video with a soundtrack) may be particularly important to the distance learner, acting as a substitute for conventional demonstration.

Music can be an expensive medium to include, since royalties must usually be paid. Alternatively, specific pieces of music might be composed, performed, digitised and edited. For the creation of IEM software in disciplines such as engineering, the skills and equipment required are unlikely to be available.

8.8.4 CAL3P's Sound Content

Sound was used minimally in CAL3P, not least because it was found that the majority of the computers which students might use during the exercise lacked a sound card. Other disadvantages which forced it to be discounted were described in Section 8.8.3.

Some of the error messages which the manufacturing simulation shows to the user when a mistake is made are accompanied by sound clips which say "Uh-uh" or "Are you sure?", but in truth these added little to the quality of the learning experience. In an IEM package on another subject, sound might be much more useful.

The only other use made of sound was where it accompanied video clips. Here, it was valuable because it added to the realism of the videos. As section 8.6.4 described, it was hoped that videos would show how some real machining operations are actually quite violent; reproducing the sounds which emanate during a metal cutting operation was a vital part of this.

With so little use being made of sound in CAL3P, it is hard to quantify any increase in the educational effectiveness brought about by its inclusion in the software.

8.9 The Value of Each Medium

Since the presentation of each medium involves a different level of development effort (and requires specific skills and equipment) it would be useful to quantify their effectiveness. This might make future development work simpler, indicating which

approaches are most worthwhile and thereby resulting in IEM software which requires less modification during production.

There are two sources of information from which the effectiveness of each of the media employed in CAL3P can be estimated; students' comments when evaluating the software subjectively, and their performance in the built-in quiz. Of course, the 'trick questions' which were used to identify experienced engineers must be discounted but each of the remaining 15 questions in the CAL3P quiz offers results. Each received a response in 32 sets of quiz results from novice engineers. Thus, performance in each aspect tested can be measured on a scale of 0–32. From these it is possible to suggest which facts and skills were easiest to learn. Since each fact or skill was taught by a different medium or mixture of media, the most effective approaches can be identified. Further, because the amount of development effort involved in the presentation of each medium is known, the most efficient media among those employed may be identified.

The subsections which follow comment on the apparent effectiveness of each medium employed (text, graphics, video, animation and sound), plus some other factors which may have influenced students' performance.

8.9.1 Results Observed with Text

It was found that text was comparatively easy to generate for use in CAL3P and plenty of suitable material was available for inclusion in the software. Obtaining other media was less simple. Existing lecture notes, slides and student exercises formed a good starting point in the prototype's development process, and typing a screenful of information was certainly faster than using another medium to represent the same information. Occasionally it was necessary to question a machinist or read up on a process in existing textbooks before writing a suitable overview for inclusion within CAL3P, but this work was never as involved as the capture of other media.

Care is still necessary with text; words must be chosen to ensure that the software is free from unnecessary jargon, and the text must be arranged to be as clear and attractive as possible. Overall, the creation of two full pages of text information per hour was found to be a comfortable pace. Where it was only necessary to enter text from an existing source, six or more pages could be entered in an hour. Decisions still have to be made, such as how to break up a long passage of text so it can be spread across multiple screens.

Observation of students using CAL3P showed that text was seldom read exhaustively. It might be interpreted that much of the development work had gone to waste, though students use of text was not dissimilar to the way a textbook is employed; few people read a technical book from cover to cover, instead referring to sections which are of current interest.

In the quiz, questions 1, 2, 4a and 10 tested for facts which were taught predominantly by text. In each case, there was also a diagram, but it was necessary to read the text to understand what was shown. Some other questions tested for skills or facts which were taught partially by text. For example, Question 6 concerned straight turning, which was the most fundamental machining process employed in CAL3P. The introductory section of the software described straight turning, teaching predominantly with text, but students had to complete an interactive mini-test to show that they understood process planning syntax before they could go on to use the rest of CAL3P. Thus, it is impossible to say if the text or the interactive test were responsible for the students' subsequent high level of proficiency in answering questions concerning straight turning. The latter seems likely, since few students were seen to get the test (shown in Figure 8.6) right first time.

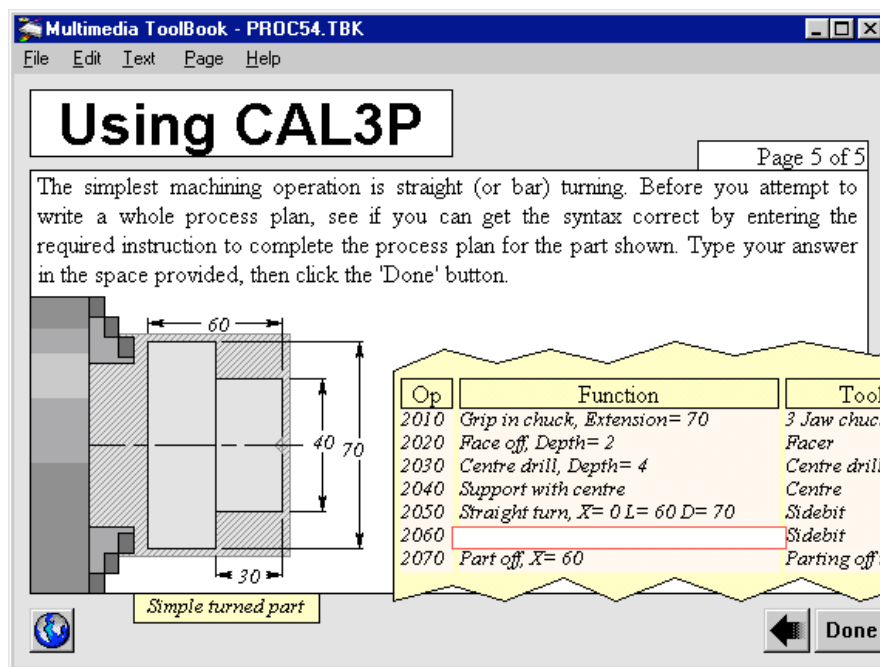


Figure 8.6: Interactive Test in the CAL3P Introduction

The three parts of question 9 were also taught predominantly by text, though the information tested was also supplied in the introductory lecture on process planning. Only questions 1, 2, 4a and 10 reliably test the effectiveness of text, since in questions 6 and 9 the answers could be learned from other 'media', as described above.

For each question, the number of correct answers received from novice students is shown in Table 8.3:

Question	Subject	Correct	Responses
1	Drilling	25	71.9%
2	Internal boring	19	59.4%
4a	Barrelling	7	21.9%
10	Taper turning	16	50%

Table 8.3: Number of correct responses from novice engineers when questioned in relation to facts taught predominantly by text

In order to identify the most effective media, the proportion of correct responses obtained must be considered against the novice students' performance as a whole. Discounting novice students' understandably poor performance on 'trick' questions, on the remainder they gave correct answers 72.4% of the time. Overall performance by classes of novice engineers ranged from 68.6% (Manufacturing Management students) to 79.6% (Manufacturing Engineering students). The worst quiz performance was by a student who scored 46.7%, and the best was by two students who each scored 100%.

Compared with an overall 72.4% mean of correct answers, none of the quiz questions which tested skills or facts acquired from text, was answered so successfully. Question 1 came very close, but it must be remembered that drilling was a process which students were required to use to create a part of the shape required by the exercise. Thus, they had an opportunity to learn about drilling interactively. It may be that some students initially misnamed the process or tried to use the wrong technique, but problems with their process plan would have been brought to their attention during the machining simulation. At the end of their time with the CAL3P software, 71.9% of novice engineers gave the correct answer.

Question 2 had no such benefit because students did not need to employ a boring operation. As such, Question 2 may provide a truer insight into the power of ordinary text, facilitating only passive learning. Poorer performance in the quiz resulted. The value of interaction in a teaching process is discussed further in Section 8.9.6.

Very poor performance in Question 4a implies that 'optional' information was read by very few students. The information which students needed to have read if they were to answer this question correctly was in a part of the software called 'A guide to turning principles' which could be accessed from within the section where process plans were written. It was thought that hyperlinking would be an effective architecture for learning; in allowing the student to choose which areas to visit, each one should correspond to one of Buzan's [1993] requirements for effective learning as "items which are of particular interest to the learner." Fisher [1994] was particularly in favour of software architecture which allowed students to explore what he called 'tangential information'.

It may be that all or most of the students who visited the guide to turning principles got the question right; CAL3P does not keep a record of which pages of information are visited. However, since performance on Question 4a was much poorer than for other questions which tested recall of information acquired from text (see Table 8.3), it must be deduced that the majority of students did not read the optional guide to turning principles. As such, development effort spent on the development of tangential information might be considered to have been wasted. The value of tangential information is discussed in Section 8.9.7.

Question 10 was answered correctly in 50% of cases, which is well below the mean performance across all questions. Students were required to select one machining instruction from a list, which would make the tapered feature shown in a diagram. The incorrect answers were chosen in approximately equal proportions. Section 3.2.3 described the memory process of coding, including how a learner may well be able to recollect the gist of a message, though he cannot repeat it word for word. This appears to have been a factor in Question 10. Students were only required to specify one taper turning operation while writing their own process plans, and it would appear that many could not recall the correct syntax for this process by the time they came to be tested in the quiz. Certainly, the alternative choices offered in this question were very close to the correct one, which makes the question much harder. In most cases, students who got this question wrong had written process plans which included an instruction for a tapered feature. It can be concluded that the students understood the principle, but could not recall the specific format required.

8.9.2 Results Observed with Graphics

Using Multimedia ToolBook as the authoring system, the graphics in CAL3P were greatly different from the HyperCard Prototype. Resolution was higher, colour graphics were much easier to display, and the system supported vector graphics in addition to bitmaps. Extensive use was made of vector graphics, as Table 8.2 showed, and some bitmaps were also included, mostly showing photographs of machine tools.

The findings of Nickerson [1965] and Haber [1970], as presented in Section 3.2.4, suggested that images were a highly effective method of communicating and memorising information. Certainly, diagrams form an important part of conventional teaching in engineering, and many had been employed in CAL3P; it was clear that graphics were the only practical way to convey concepts of size, shape and surface finish.

It remained necessary to measure the educational effectiveness of CAL3P's graphics. Questions 3a, 3b and 3c tested for information which was provided in graphical form, though graphics also played a supporting part in some other questions, as detailed in

Section 8.9.1. Table 8.4 shows the results, which should be compared with the overall mean performance of 72.4% correct answers.

Question	Subject	Correct	Responses
3a	The chuck	28	87.5%
3b	The saddle	19	59.4%
3c	The tailstock	21	65.6%

Table 8.4: Number of correct responses from novice engineers when questioned in relation to facts taught predominantly by graphics

Performance by novice students in the CAL3P quiz implies that graphics were a more effective medium than text, as learning theory had suggested. However, despite the clear value of images, simply making them available for viewing was not the most powerful way to facilitate learning. Powerful though images are, they still require only passive attention. In requiring learners to select the manufacturing processes to use, the interactive exercise required the learner to evaluate the screen displays presented, to decide if a given process would be useful to them. This matches Seale's [1997] requirement for reflective learning, described in Section 3.4.1, and the approach appears to have been more effective. Question 3a tested whether students could correctly name the chuck, a piece of equipment which they were certain to have used when writing their own process plans. To a lesser extent, this is also true of the tailstock, which was the subject of question 3c. Students were not required to name the tailstock while they wrote their process plans, but it would have been seen in any screen relating to centre drilling, drilling or supporting the workpiece with a centre. Effectively, rather than simply attempting to memorise a large number of facts about machining processes, the students had worked with the information they discovered. Thus, learning took place in accordance with Ausubel's [1968] requirement that facts which are to be learned must have a meaningful context.

Several other quiz questions tested students' knowledge of information which was taught partially by diagrams, but it is difficult to identify the exact source of a student's answer in these cases. For example, the three parts of Question 9 test students' knowledge of the influences which a process planner must appreciate. This information was provided in the form of a diagram on the second page of CAL3P's 'Guide to Process Planning', but was also made clear to the students during an introductory lecture. Table 8.5 shows the novice students' performance:

Question	Subject	Correct	Responses
9a	Part drawings	30	93.8%
9b	Estimated demand	23	71.9%
9c	Factory capabilities	30	93.8%

Table 8.5: Number of correct responses from novice engineers to the three parts of Question 9

Performance on these questions was excellent, which at first glance implies that conventional instruction is still one of the most powerful teaching methods available. However, it is interesting to note that the Manufacturing Systems Design students from B.I.H.E., who received no introductory lecture, all scored 100% on these questions. The ‘Guide to Process Planning’ was originally intended simply as a ‘safety net’ to allow latecomers or absentees to take part in the CAL3P exercise, but it appears to have been highly effective on a stand-alone basis. Indeed, it may be that when some of the other students were presented with information which was already familiar, they did not attempt to read it and thus reinforce their understanding.

8.9.3 Results Observed with Video

Video was employed in Question 7, though technical difficulties meant it could not be used with two of the groups tested. The question took the form of a video clip in which a parting off operation was shown, and students were required to name the process. Students should have employed this operation during the process planning exercise.

Of the fifteen novice engineers who were able to view the video and attempt to answer this question, nine answered it correctly. This does not represent a tremendous endorsement of the CAL3P learning process as it suggests that 40% of students could not relate the theory which had been taught with diagrams to ‘real life’ as shown in a movie clip. However, the video clip is of poor quality; technical restrictions limited the video window size and the frame rate. As a result, the video is hard to interpret. One of the students identified as an experienced engineer also answered this question incorrectly.

Other video clips, within the main part of CAL3P rather than the quiz, supported learning by showing demonstrations of generic machining processes such as facing off, drilling, boring and straight turning. Questions relating to these operations were answered correctly in a majority of cases, as Table 8.6 shows, though interactivity appears to have made a much greater impression than the actual video clips themselves, as discussed in Section 8.9.6.

Question	Subject	Correct Responses	Learned interactively?
1	Drilling	25 71.9%	yes
2	Internal boring	19 59.4%	no
6a	Straight turning	27 84.4%	yes
6b		26 81.3%	yes
6c		27 84.4%	yes
6d		28 87.5%	yes

Table 8.6: Quiz performance on questions where information was available on video

Since the computers used for two of the four software trials could not display digital video at all, it is possible to break down the results presented in table 8.6 and compare the performance of students who were able to view the video clips with those who were not. Table 8.7 shows the result:

Question	Subject	Correct Responses (able to view video)	Correct Responses (no video)
1	Drilling	11/15 73.3%	14/17 82.4%
2	Internal boring	8/15 53.3%	11/17 64.7%
6a	Straight turning	12/15 80.0%	15/17 88.2%
6b		12/15 80.0%	14/17 82.4%
6c		11/15 73.3%	15/17 88.2%
6d		12/15 80.0%	16/17 94.1%

Table 8.7: Comparison of quiz performance by students able and unable to display digital video on their computers

It was surprising to learn that the novice students who could display digital videos had not outperformed the novice students whose computers could not employ the medium. The video-capable students' performance was substantially poorer in most of the areas where video supported instruction.

Of course, as Section 8.6.4 described, the digital video clips used in CAL3P were of limited quality, but it had nonetheless been expected that there should be some increase in student performance where video was employed. It is, after all, one of the principal arguments made in favour of the use of multimedia in education that vivid media should get a learner's attention and make experiences more memorable. Section 8.6 included comments to this effect by Heath [1996], Vaughan [1994] and others.

From the results presented in Table 8.7 it appears that the opposite may be true; that the use of digital video can, under some circumstances, actually be misleading or distracting. It may be that the students who were able to display digital video were

seduced into paying attention only to the more vivid media, disregarding the large volume of useful information which appeared in text form.

The data discussed here only became available by chance, simply because one of the computer labs used during trials was not set up to allow the display of digital video. In order to evaluate conclusively the effectiveness of video it would be necessary to conduct a new experiment with a larger number of participants. For now, it can only be said that the findings to date are at odds with the statistics quoted by Leonard [1994], that we remember roughly 10% of what we read, 20% of what we hear, 30% of what we see, and 50% of what we see and hear. Students who had only read text and studied diagrams were better able to answer questions relating to machining processes than students who also had the additional opportunity to see and hear manufacturing operations.

8.9.4 Results Observed with Animation

As Section 8.7 described, CAL3P contains just one animation, albeit a very complex one which can show a range of different turning processes. The development effort ratio for this is hard to pinpoint because the animated machining sequence is likely to be viewed repeatedly while the student experiments with the machining processes which are available.

In the on-line quiz, a general increase was seen in student performance on questions which tested for information which had been learned or reinforced through use of the animated machining simulation. This provides the best indication of a DCG animation's educational effectiveness. The machining simulation was also popular with students, according to their software evaluation forms, the results from which were summarised in Section 7.6.

8.9.5 Results Observed with Sound

Sound remained largely unused in CAL3P, not least because of the drawbacks associated with the medium when employed in a classroom environment, as described by Travis and Jones [1997] (see Section 8.8.3). Also, investigation had shown that most of the computer labs where CAL3P might have been used did not boast a sound capability, so effort expended on providing information in this medium could have been wasted in many instances.

It is not strictly true that an application must incorporate every medium if it is to be described as multimedia. Each medium should be employed wherever it would be the most appropriate way to convey a piece of information. As a piece of software about machining processes, CAL3P was heavily dependent on animations and diagrams; a piece

of IEM software about composers would employ an entirely different mixture of media, and would certainly make much more extensive use of sound.

8.9.6 Results Observed with Interactive Learning

Sections 8.9.1 to 8.9.3 gave student quiz results, and in each case it was noted that learning had been more effective where the information had been provided in a manner which allowed interaction. For example, Section 8.9.1 described how the first quiz question had been answered correctly in more cases than the second, which referred to a machining process with which most students had probably not experimented when writing their own process plans.

This is congruent with the learning theory presented in Chapter 3; that information which requires manipulation and evaluation is more likely to be recalled accurately. The cost of making IEM software highly interactive, in terms of development effort, is substantial. Section 8.7.4 described the programming work undertaken in providing a DCG animation, indicating a development to playback ratio of 3600:1. However, learning was seen to have been most effective where interactivity was highest. With regard to the other major constraint under which IEM software development takes place, it was found that allowing interaction did not impact significantly on the technical requirements of the delivery system.

8.9.7 Software Architecture, and Learning from Tangential Information

While it can take considerable time to compose the text for a CAL package, generate diagrams, take photographs etc., there is no guarantee that a given page of information will necessarily be read by the user, much less learned; some people will skip past the page in search of other information. This is also possible with a book, but it is potentially a much greater problem in an IEM package which employs hypermedia. Where a piece of software features numerous hyperlinks, learners can easily find themselves lured off topic. Even in a very precisely targeted piece of IEM software where every page of information is potentially useful to the user (as CAL3P should be for a novice process planner), non-linear access may be less than ideal because it can be very difficult for the learner to ensure that they have read everything which might have been useful.

Fisher [1994] suggested that one of the advantages of a non-linear document is that by selecting topics which are of interest or current relevance, the user is more likely to learn effectively. However, quiz results were poor on questions which related to information made available via optional hyperlinks. Question 3b was answered correctly in 59.4% of cases, and Question 4a in only 21.9% of cases, compared with the overall mean performance of 72.4%. It may be that tangential access to information resulted in a very

effective learning process, but that many students did not visit the section which offered the information. It should be borne in mind that with the exception of the M.Sc. students who used CAL3P in December 1997, software trials were not subject to a time limit, so there was no reason for a student to neglect to visit a section – unless they were unaware of its existence. This is doubly unfortunate, not only because it shows that some students failed to learn certain facts, but also because development effort (and disk storage) had been committed to the provision of material which many students had not used.

As a result, it appears that a hyperlinked architecture was a less effective form of IEM than some of the other methods employed. It may be that referring to something repeatedly but in a subtle manner, such as the way the tailstock was frequently mentioned in passing, was a more effective teaching method than formally offering the student an opportunity to go to a section devoted to the subject. This is supported to some extent by learning theory, with Ausubel's [1968] contention that information is more likely to be memorised when it is given meaningful associations with existing knowledge. Following a hyperlink to a whole new section may not allow these associations to form so readily.

While hypermedia can allow rapid access to new sections, it can leave the user struggling to build up a mental image of the structure of the software. Each move tends to take the student deeper, to more specific information, but in a large hypermedia environment it can be hard to ensure that every link has been explored; that everything has been read. If a page offers several exits, it can be hard to decide which route to follow. Getting lost within the information becomes a problem. One solution is what Leonard [1994] calls a 'running button'; a button which can be found on every screen, leading back 'home' to the top level. CAL3P's 'globe' button is an example. From here, the user could then retrace their steps to the screen which offered several choices, and select another. Used in this manner, hypermedia seems inefficient. The user may be left with the suspicion that he missed whole sections each time a page offered multiple paths for him to follow. This is an equally unattractive prospect for the developer of a piece of hypermedia IEM software, since development effort is essentially wasted on sections of information which are seldom accessed.

8.10 Conclusions

It had been hoped to simplify future multimedia development somewhat by identifying which medium or combinations of media could be expected to facilitate the most effective learning. This is necessary because limitations exist for any IEM authoring project. Section 8.1 classified these as limitations of a technical nature (relating to the

specification of the end user's computer), of users' willingness to make use of the software, and of development cost. Operating within these limitations, an understanding of the peculiarities of each medium should allow the best possible use of the resources available.

Using results from CAL3P's on-line quiz, it was hoped to make a calculation based on the resource used to present a piece of information (in terms of development time or technical implications) and the level of student attainment observed as a result. It was found that some media appeared to have been more effective than others, and that the use of each involved a measurable amount of work, and measurable technical requirements. However, all experimental results were influenced by the context in which the media were presented, not least the degree of relevance to the exercise which students were working to complete.

Ultimately, what had begun as an effort to quantify the value of each medium, so far as possible, revealed that the most effective learning was facilitated not by any one medium, nor even any specific mixture of media, but by allowing the user to interact with the software. This is an important finding which may lead to the creation of more effective IEM software in the future, but the unexpected degree to which the level of interactivity influenced quiz results means that experimental data intended to measure the effectiveness of individual media are biased to some degree. It would be unrealistic to expect any real-world subject to allow an entirely fair test of IEM software; the difficulty of individual tasks within the learning process will vary, the developer or development team's proficiency with each medium will be different, students' existing level of knowledge in a subject may be hard to predict, and certain facts may be more applicable to being taught with certain media than others.

The only reasonable way to offset these influences was to describe the learning situation and the software in detail, such that the reader might judge where the software has been most effective. High quiz scores do not indicate excellence in the software if the students were already familiar with the material, if the skill which was tested is very easy to acquire, or if the alternative answers to a multiple choice question were clearly nonsensical. It has been necessary to discuss in detail the results obtained, in some cases qualifying the results obtained, such as where poor quiz marks may indicate that hypermedia sections were not accessed, rather than where they were seen but not remembered.

8.10.1 Correlation Between Learning Theory and Results Observed

Certainly, the learning theory summarised in Chapter 3 was borne out by the novice engineers' quiz results. Scores suggest that the most effective learning took place where

the student was required to manipulate and apply the information rather than simply trying to read and remember (in accordance with Hunter [1957]). Equally, in determining which machining processes to employ, and when deciding if the result was satisfactory, the learner had to employ reflection, as demanded by Seale [1997]. Sections 8.9.1 and 8.9.3 both describe quiz results where students' performance was better on questions where reflection had been required.

The software's most effective aspect appears to have been the animated machining simulation, which students visited repeatedly as their process plans evolved. Here, the software allows learning to take place as a cyclic or iterative process, as suggested by Kolb [1974], Crampes [1991] and Yolles [1997].

Within the learning cycle, CAL3P's machining simulation appears to have functioned as a reinforcer. Feedback is an essential part of the learning process, if only to tell the learner that he has correctly interpreted the material presented, or to require that a section has not been mastered. However, simple right/wrong feedback at the end of the session – such as the quiz provided – cannot really motivate the learner. Reinforcement requires that the learner is given a reward (and negative reinforcement requires that the learner is punished).

Viewing the machining simulation being carried out became a reward when, step by step, it produced features of the required geometry and ultimately a completed workpiece. Equally, the learner was 'punished' when an erroneous instruction ruined the workpiece, or caused the machining process to end prematurely because of an instruction which would damage the machine tool or place the operator in danger. In this way, errors which students encountered were much more poignant because they related directly to their own work. Few pieces of IEM software can offer such effective reinforcement (see Figure 8.7), though a reward of some kind is a key element of Gagné's [1975] learning model.



Figure 8.7: Tangible reward and punishment in computer-aided learning: this never happens!

Images were seen to be a powerful way of communicating with learners, with subjects supported by diagrams apparently resulting in a higher level of student performance than those taught purely by text, for example. Again, learning theory had predicted this. Nickerson [1965] and Haber [1970] had both found that humans have a tremendous capacity to recall images (see Section 3.2.4). Of course, the on-line quiz did not simply require that students distinguish between images as seen and previously unseen, so performance in graphic-based sections of the quiz was not as good as that observed by Nickerson or Haber. CAL3P's diagrams and photographs supported learning of engineering principles, and answering quiz questions required that these principles should be applied. Nonetheless, facts which were supported with relevant visual media appear to have been more reliably learned.

It may also be significant that CAL3P was well received by students. Results from the evaluation forms, presented in Section 7.6, indicated that the majority of students felt the exercise was both more enjoyable and more worthwhile than a conventional tutorial. This can only have improved the level of attention which the students applied during the exercise; motivation is recognised as a key factor in successful learning, such as in Gagné's [1975] learning model, and Wheldall's [1983] study of motivation and workload.

8.10.2 CAL3P's Most Efficient and Effective Media

As Section 8.10.1 described, images appear to have been the most effective medium, in the strictest sense of the word. Facts which were taught with the support of images were more often recalled correctly during the quiz. However, the greatest single influence

on the educational effectiveness of CAL3P's modules came not from the use of a particular medium but from the level of interactivity allowed. As Section 8.9.6 described, sections where the student was able to experiment resulted in consistently better quiz performance than those which supported only passive learning.

To some degree, interactivity can be subjected to the same measurements as the individual media which make up IEM software. With greater interactivity comes an increase in the complexity of the software, possibly requiring higher specifications in the hardware meant to run the software, and certainly involving additional coding work. In creating software which allows the user to experiment, additional effort is invested on the basis that the software will be more effective as a result.

Unfortunately, the degree of interactivity is difficult to measure. As Section 4.6 described, the higher levels in Schwier and Misanchuk's [1993] classification of interactivity focussed almost entirely on technical capabilities such as touch screen technology and speech recognition – equipment which can make communication with computers more intuitive, but which does not define the flexibility of the software itself. While the number of words or photographs available in a piece of software can be counted, the level of interactivity is a more abstract concept, not least because the code which supports simulations and other interactive elements is hidden from the typical user.

Building on Boehm's [1981] Constructive Cost Model, Marshall et al [1994] offered the Multimedia Effort Estimation Model (MEEM). This attempts to predict the total number of man hours which will be involved in an IEM development, given typical playback time and assessments of the complexity of the software and the circumstances under which its development would take place. Within the MEEM, measures of software complexity are called the interactivity cost drivers, though only four of these could really be said to concern the level of interaction with the student. These are the complexity of the interface, the level of activity (user's control over branching and navigation), quiz question style and the type of question feedback. The remaining interactivity cost drivers address the 'density' (mean quantity per page) and potential source of non-text media such as movie clips, it being assumed that text is the default constituent of each page.

The CAL3P software does not fit perfectly into the MEEM. For example, should CAL3P's animated machining sequences be considered as an animation, a simulation or both? Since all machining simulation is carried out on a single page, the density of animation/simulation would appear to be very low, which in the MEEM implies that little work is required. If each different function which can be shown in the machining simulation is considered a separate animation, the density value is very high. Thus, CAL3P could occupy virtually any position on the MEEM scale. (Nonetheless, Marshall et

al [1994] made a number of valuable points which should be addressed when budgeting for an IEM authoring project, taking into account factors such as reuse of existing course material.)

In addition to investigating which media were most effective, it is possible to grade them in terms of efficiency. While the approaches of presenting information with graphics (and particularly when incorporating interaction) were seen to produce the best performance in the on-line quiz, these are not necessarily the most efficient techniques to use. As Section 8.5.3 described, an image can take up much more computer storage space than text conveying the same message. Graphics can also be much harder to create or obtain, so it is possible that technical limitations or constraints on the development process may necessitate the use of a less effective but less 'expensive' approach.

For CAL3P, plain text was found to be the most economical medium in which to provide information, requiring far less storage space and development effort than other media. It is likely that this will be true for most IEM applications, though of course there are instances where text is wholly inapplicable. Among the students tested by the CAL3P quiz, the proportion of correct answers received was approximately 15% lower on questions which related to information presented as text; this may be considered an acceptable drop in performance, given that a text-based application takes less time to create. IEM software is seldom created out of a desire for economy, since savings are only likely where the target audience is large (such as the American Airlines case presented in Section 4.7.1). Generally, the aim is to facilitate superior teaching – something which will not be achieved by a simple conversion of existing text-based training materials. Equally, no single medium can be used exclusively if a piece of software is to be described as multimedia, and too great a reliance on any single approach would probably reduce its impact and adversely affect the quality of the learning process.

One medium was found to be neither efficient nor effective; the effort expended to include video clips in CAL3P appears to have been largely wasted. As Section 8.6.4 described, digital video files took up a massive amount of storage space, changing CAL3P from an application which could be installed from a few floppy disks to one which required a CD-writer to be employed each time an update was to be issued. The large size of the software also precludes it from being offered for downloading via the Internet. Despite increasing the size and complexity of the software, trials suggested that poorer student performance resulted when video was available. Section 8.9.3 showed how students who were able to see video clips actually scored less in the on-line quiz. It seems foolish to spend a substantial amount of time (Section 8.6.4 gave the development effort ratio for CAL3P's video content at 382:1) on a production activity which increases the

technical requirements of the playback system – and at the same time apparently reduces the quality of the learning experience.

There is a danger, as Goble [1994] warned, that multimedia is thought of in terms of technology rather than application; Heath [1996] considered digital video a vital constituent of multimedia (see Section 8.6) but results from this research suggest there is no guarantee that a medium will be more effective simply because it has greater technical requirements. Indeed, results obtained with CAL3P suggest that an increase in the educational effectiveness of IEM software can be accompanied with economies in the hardware specification required for playback.

The ninth chapter summarises the findings of this research and presents some suggestions for further work.

9.0 Overall Discussion

Following on from the review of experimental results presented in the previous chapter, an overall discussion of the way forward for multimedia in education is offered, starting with a summary of the work which was conducted.

9.1 Summary of Work Conducted

This thesis has detailed an investigation into the applicability of multimedia technology to engineering education. It was identified that multimedia technology has excellent potential for reaching new learners, and for improving some of the instruction given to people currently engaged in learning. Unfortunately, problems were found to exist; educational software – particularly that which makes use of multiple media – is time-consuming and expensive to develop. Differences in teaching style and subject emphasis have created a culture where few pieces of software receive sufficiently widespread use to justify the development effort invested. There is also a danger that the creation of multimedia software may be technology-led rather than user-centred, with little attention being paid to established educational psychology.

Opinions presented in Section 3.6 suggested that the creation of CAL (and therefore IEM) software is often undertaken with enthusiasm but little concrete knowledge of the best approach to take. Section 8.1.3 quoted Bhide and Nene [1997], who suggested that any good teacher should be able to produce CAL software. To take this approach when creating educational software is to court disaster, given the massive amount of work involved and the tremendous range of options available to the developer. It was considered desirable to reach a better understanding of how development effort might best be expended.

Towards this end, learning theory was reviewed to identify how the findings of educational psychologists could be applied to CAL and IEM software. Models of the learning process were presented which showed that it is essentially cyclic, consisting of distinct phases. Each phase requires different actions from the learner (or teacher). In Section 3.6, warnings that much educational software design was not based upon learning theory were noted. The alternatives identified were to make design decisions on the basis of computer hardware limitations, or out of expediency in development – neither of which guarantees a profitable learning exercise.

To demonstrate how these limitations might be overcome, several pieces of IEM were written, demonstrating features which might be desirable in future software. The

most significant of these was 'CAL3P' (Computer Aided Learning of Process Planning Principles), a training tool for novice engineers.

9.2 Conformance to Project Aims

The aims of the project were presented in Section 2.1. They may be summarised thus:

- To test, through the development of a new piece of software, the assumption that CAL / IEM provide instruction which is better, cheaper or more accessible than the conventional teaching methods.
- To provide a useful educational experience for test subjects.
- To identify the features of an IEM application which make the most substantial contribution to the learning process.
- To contrast the effectiveness of each medium with the cost inherent in its capture and presentation.
- To demonstrate alternatives to the conventional approach to multimedia software, perhaps producing multimedia feedback 'on the fly', in response to user inputs.
- To push interactivity as far as possible.
- To demonstrate how multimedia could be used within an interactive tool rather than in a simple one-way training process.
- To investigate the applicability of learning theory in a computer-based learning environment.

The creation of the IEM application 'CAL3P' addressed the first and second aims. Section 4.2 included details of several CAL applications which had been described as successful, and CAL3P was no exception. It addressed a subject which was difficult to teach by conventional methods, as described in Section 6.2. The same chapter concluded that the process was improved substantially, for both learner and teacher.

Section 7.5 gave details of the overwhelmingly positive response to CAL3P by students, identifying the software as more worthwhile and more enjoyable than a normal lesson, showing that the second aim was achieved.

With regard to the identification of key characteristics of IEM software, Section 8.9 discussed the experimental results obtained, and Section 8.10 concluded that interactive elements were the most effective means by which people might be made to learn. It was

suggested that for this reason, interaction should be considered a medium in its own right during the software planning stage. (Sections 8.4 to 8.8 had discussed the nature of each medium, their merits, the opportunities they present, and their potential disadvantages.)

The fourth aim was to contrast the effectiveness of each medium with the cost inherent in its provision. If development effort (and cost) were to be taken into account, as Section 8.9 describes, then interactive elements would not appear viable. (Section 8.9.6 put the development effort ratio for an interactive animation at 3600:1, when a figure of 300:1 is normally considered prohibitive.) In purely economic terms, text is the best medium. A higher proportion of learners will remember a fact which they learn in an interactive process, but the costs involved far outweigh the performance benefit observed; interactive elements must be confined to key skills and facts, unless for some reason development cost is not an issue.

Both CAL3P and the DICTA Information system (described in Chapter 5) demonstrated that a multimedia system could generate specific information on demand, in response to a user's inputs. As such, they addressed the next three aims of the research, showing how a system which incorporated multimedia might be used regularly, even daily, as opposed to being used merely for periodic training. Whereas most IEM software can be considered a static 'document' to be read through just once or twice, the DIS offered dynamically generated content, created to represent the design of a printed circuit board. It could be worthwhile to consult such a system repeatedly, receiving different comments based on the design created in the CAD system.

CAL3P was programmed to generate animated sequences which represented a machine tool being operated in accordance with the student's instructions. A conventional approach to such an IEM project might have employed a number of digital video clips showing generic machining processes; CAL3P surpassed this, allowing students to experiment. To use multimedia in this way was highly unconventional; descriptions of the way CAL3P operated were met with disbelief from authoring software vendors on more than one occasion.

Finally, the experiments which were conducted made it possible to assess the applicability of learning theory in a computer-based situation. Section 8.10.1 discusses this, concluding that the learning theories presented in the third chapter were borne out, both by experimental results and qualitative feedback. With regard to the research objectives identified in Section 3.7, the most significant points are:

-
- Dynamically generated content – screen displays which represent the student’s own input – were seen to provide an adequate reward, motivating the learners.
 - Coding – the manipulation of information in short-term memory such that it can be stored – was found to take place reasonably well. The capacity of short-term memory is limited, however, (see Section 3.2.4) so the learner must not be presented with a complex user interface. When new functions must be added to the user interface, there should be an introductory screen where the student can master the interface without having to learn new subject material. Figure 7.4 shows an example.
 - As had been expected, images were found to be tremendously effective, through the influence of graphics within the software was outweighed by that exerted by the interactive machining simulation – a feature which could also be considered to make use of images, though they were animated. The tremendous image recall performances found by Nickerson [1965] and Haber [1970] were not enjoyed, probably because CAL3P’s test subjects were required to do more than simply recognise an image as seen or previously unseen; CAL3P students had to learn a skill and then apply it.
 - CAL3P’s built-in quiz, with its special ‘trick questions’ (see Section 7.1.1) to detect learners who had substantial prior knowledge of the subject, demonstrated how it is possible to measure the amount of learning which takes place while a piece of software was used.
 - Learning theory was found to be valid within the IEM context. In particular, activities of an iterative or cyclic nature (see Section 3.3) were found to be a highly effective teaching mechanism.

9.3 Recommendations

In Section 4.7.3, and subsequently in greater detail in Section 8.1, the limitations to the proliferation of IEM software were identified. They can be classified into three broad groups; those relating to users’ acceptance of the technology, those concerning development effort, and those impacting upon technical constraints.

There is no simple way to identify which of these limitations is the most important. In fact, there is a subtle inter-relationship between the three. The developer who specifies a very simple IEM package will save development time and money, but the results seen when using such software may discourage further development. Even if successful in the

institution where it was created, the software may well be too narrowly specialised to be acceptable elsewhere (the “my material for my kids” approach identified by Turnbull [1974], as described in Section 8.1.3). Equally, a piece of software which is designed to be suitable for a wide variety of different users may be so complex that it proves too expensive to complete, or it may exceed the storage space available on the delivery system.

The CAL3P software created during this research demonstrates how the limitations under which educational multimedia must operate can be ameliorated. Sections 9.3.1 to 9.3.3 address users’ acceptance, development effort and technical limitations respectively, suggesting how these problems might be overcome.

9.3.1 Achieving Acceptance for IEM Software

CAL3P is a flexible piece of software which students can use to plan the machining of an almost infinite variety of workpieces. Thus, a tutor could incorporate the software as an exercise at GCSE level, or in the teaching of engineering fundamentals on a degree course. Because the tutor can determine the degree of difficulty simply by giving the students a different part drawing, CAL3P is likely to be considered suitable in more situations than a normal IEM program with a ‘hardwired’, built-in exercise.

CAL3P’s machining simulation encourages experimentation, an approach which students reported that they enjoyed. This method matches the cyclic model of learning (see Section 3.3), demanding that students apply the information which is presented to them rather than simply memorising it. The iterative nature of the simulation makes it work as a reinforcer, maintaining most students’ interest and motivating them to master the system. Few students reported dissatisfaction. Thus, by incorporating high levels of flexibility and interactivity, the likelihood of users and tutors finding the software acceptable was improved substantially. For a commercial IEM package, responses such as CAL3P received would be highly desirable, since reaching a large number of users would allow the recovery of development expenses.

9.3.2 Operating Within Technical Limitations

Displaying the most attractive, attention-grabbing media can be problematic. It might be felt that photo-realistic images and broadcast quality video are desirable since they are likely to improve a potential user’s reaction to a piece of software, but such media require the storage of a large amount of data, to be accessed at speed when required. Writing IEM software which makes extensive use of ‘rich’ media may actually reduce the size of the audience, as users with slightly out of date computers are put off by delays and jerky video clips.

In the past, when computers' capability to display rich media was limited, programmers still managed to create some highly effective educational software. Chapter 4 identified Logo as one such program, for example. Although Logo could not display attention-grabbing media, it endeared itself to users by offering considerable flexibility. Allowing the learner to enter instructions – to 'play' with the software – meant that it could be used meaningfully for much longer than a less interactive piece. Low technical requirements, simplicity and flexibility meant that Logo enjoyed widespread use.

Chapter 8 detailed the relatively massive amounts of storage space which media such as video clips and high-quality audio require. Using these can reduce the flexibility of an educational multimedia application. Even when using CD-Rom to store the program, the space available can rapidly be filled by these files, and the cost to create or acquire these media is also an issue. Thus, there is a limitation on the number of rich media clips which can be incorporated, and as a result there is a very real temptation to write multimedia software which is overly prescriptive; by limiting the learner's freedom, the author can ensure that each of the media which has cost so much to incorporate (in terms of money or storage space) is played. The result is a linear multimedia program which may not make the best impression on a learner. Section 8.9.6 discussed results from student attainment tests, in which it was found that learning had occurred most effectively where CAL3P gave learners the freedom to experiment to a much greater degree than IEM software would normally allow.

9.3.3 Justifying Development Effort

As Section 4.7 described, the effort involved in IEM development can be prohibitive, possibly requiring hundreds of hours' work to 'computerise' a single tutorial. It was explained that development effort is typically measured as a ratio of hours spent in the creation of a program against the time which a typical student might spend using the software. Similar measurements were taken during the creation of CAL3P for each of the sections of information within the program, and the media used.

The amount of effort involved in putting media such as photos and video clips in CAL3P was typical. The most significant departure from the commonly accepted development effort ratio of 300:1 was in CAL3P's animated machining simulation. Section 8.7.4 discussed the development effort for this, arguing that the figure of 3600:1 for a single playback was misleading, because the animation would be viewed many times during a process planning exercise.

When an IEM program is geared towards retrieving and presenting predefined content, as is usually the case, the software can only be of real value to a learner a very few times. Repeating the same exercise over and over again is of diminishing value.

Section 4.3.1 contended that revisiting a section of information until a pass mark is obtained in a quiz is unlikely to produce real learning, because the student is simply looking up facts in order to meet a short-term goal. The CAL3P software can meaningfully be used repeatedly, perhaps writing process plans for parts of increasing complexity or investigating alternative process plans for an existing part. Thus, CAL3P is better suited to repeated use than a conventional IEM application (one with a purely prerecorded store of media), and it becomes easier to justify the time expended on its development.

The DIS software described in Chapter 5 could also be used regularly, perhaps each time a design engineer produced a new PCB layout. Eventually, the multimedia sections would decline in importance because the engineer would be familiar with the advice which they give, but the alert issued when a component is in violation of a design guideline would still be valuable.

9.3.4 Further Recommendations

Good working practices for the development of IEM software have been identified in many sections of this thesis. For example, Chapter 3 reviewed theories of how the human brain works, identifying important factors such as motivation and reinforcement. Sections 8.4.1, 8.5.1, 8.6.1, 8.7.1 and 8.8.1 described how the quality of each medium might be maximised by following established principles. Some of those guidelines originated within the relevant media industries such as print, film and music; the remainder are specific to digital multimedia, arising because of the limitations of computer hardware. This combination of guidelines from various sources is an important part of this thesis, which could save a novice developer a great deal of time and expense.

Finally, one of the principles which Horton [1994] suggested for on-line documentation in general (some of Horton's recommendations appear in Section 8.4.1) is particularly pertinent to IEM development; make fewer presumptions about the user. The user's level of interest might range from thoroughly absorbed to extremely reluctant to learn. Similarly, their purpose in using the software might vary, as might their level of prior knowledge. A user might be an expert simply seeking rapid confirmation of a single fact, or he might be a novice, seeking a full lesson but being unfamiliar with the terminology used. Only a piece of software which can address the needs of such an apparently diverse set of individuals will be widely reported as useful and enjoyable.

9.4 Further Work

In the course of this work, several areas have been identified in which future research would be beneficial, in order to further address the shortcomings of a computer-based learning environment and to capitalise on its unique strengths. They are presented in the subsections that follow.

9.4.1 Body-language and the Computer-based Tutor

Despite the flexibility exhibited by CAL3P, a human tutor may provide superior instruction if he is able to adapt the lesson being delivered, taking account of students' reactions. Where CAL software must present information in a largely one-way process, a human tutor can interpret the body language of students. He has the option of changing his delivery pace, going over problematic areas again and selecting examples which the current group of students are likely to find familiar. It might be thought that CAL will always struggle to achieve these things adequately. Certainly, it is true that the computer does not take heed when a user frowns or scratches their head. One form of 'body language' which might be measured, however, is the timing of a student's inputs. This might be useful in determining when a learner needs assistance. The answer to a single quiz question does not mean a great deal, particularly with multiple choice answers which may simply have been guessed. However, using a suitably designed quiz which takes note of the time a student takes to supply an answer, it may become possible to identify gaps in understanding; a correct answer which is given only after a lengthy pause might suggest that the student is struggling, and likewise a rapidly given, incorrect entry might indicate that a false assumption is being made. From this information, feedback and further exercises might be tailored specifically to address a student's apparent failings.

9.4.2 Towards a Better Understanding of the Effectiveness of each Medium

Although results from CAL3P's built-in test have identified which media were the most efficient and effective, within the context of a process planning application for novice engineers, it was not possible to quantify accurately the value of each approach in relation to the costs inherent in its provision. This difficulty arose principally because it was necessary to teach a part of an existing subject; this meant that some media would be better suited to the subject matter than others, and that some students were already familiar with some part of the exercise. For a more detailed examination of the effectiveness of each medium, a topic would have to be chosen of which the test subjects could have no prior knowledge (Section 2.11 suggested that a fictional scenario might be adopted). Since a large number of test subjects would be preferable, the Internet might be

used to reach a large number of volunteers, who would play some kind of on-line 'memory game'. Their performance would be used to measure the efficiency of a variety of media and presentational styles. Ultimately, it might be possible to construct an IEM development model where desired learning outcomes could be cross-referenced with the most appropriate teaching style, the mixture of media which should be used, and the development time which should be expected.

9.4.3 Adaptive IEM Courses

A great strength of the computer in education is that it is tireless and can be available to deliver material 24 hours a day. If students can determine when they will receive instruction, why not how? Instead of developing a piece of software and then polling students for responses after they have finished using it, it might be beneficial to ask each user how they would like to be taught, and then have the software display the appropriate information.

The user might be required to choose between a number of statements, such as "I like to be shown a demonstration before I try something for myself," or "I prefer real-life case examples to pure theory." As section 4.5 showed, allowing just four such options could lead to there being sixteen different routes through the software. Where choices were offered, development effort would double (assuming each preference question offers only two choices), but this added flexibility might produce better learning. Buzan [1993] contended that items which were of particular relevance to the learner were more likely to be learned successfully (see Section 3.4.3); perhaps an adaptive system would match the learner's desires more closely, producing more peaks in attention level and more rigorous learning as a result.

9.4.4 Closer Activity Monitoring

One problem for computer-based learning may be that it ultimately becomes too user-friendly and accessible. Why should a student be motivated learn now, if the program will still be available to teach them later in the week, or next month, or on the eve of their examinations... The computer cannot command the respect of the learner, so it may be used in an offhand manner.

An experiment undertaken during this research showed that the computer can be programmed to take on some of the supervisory workload. A short IEM programme on the subject of casting metals was created in which the amount of time the user spent on each page was recorded in a database. When the user was subsequently required to pass a test, the computer gave feedback based upon the amount of time the student had spent on the relevant page. Those who had skimmed through pages of information without taking

time to read them and then failed the test were criticised and offered the chance to go back and find the right answers, but a confident user was still free to advance rapidly through the material. In effect, people with several different learning styles and levels of ability could all receive different feedback.

A more detailed investigation of closer learner activity monitoring would involve a more accurate assessment of the student, built up throughout the lesson, or over several sessions. There would be a larger set of responses available to the computerised tutor, driven by more subtle methods, such as the actual answers given, rather than simply the number of correct answers.

9.4.5 An Investigation of how IEM Software is Used

When a completed piece of IEM software is released for use at other institutions, the authors may be surprised to learn how it is used elsewhere. This was demonstrated in the case of CAL3P; in the trial which took place at Bolton Institute of Higher Education, the students' exercise was not complete once the quiz had been finished. Once everyone had completed the exercise they restarted the program and demonstrated their machining simulations to each other, defending the choices they had made.

If the development effort involved in creating IEM software is to be justified, it is desirable for the end product to be useful to as many people as possible, but experiences with CAL3P suggest it is not known how an IEM application will actually be used. Had it been known that the software might be used as it was at B.I.H.E., it would have been very simple to add a screen at the end of the quiz, offering an opportunity to display the machining simulation again.

There may be as many different ways to use a piece of IEM software as there are educators. Widespread trials and consultation with teachers would establish more precisely what is sought from multimedia. By putting the emphasis on educators to state what they want – rather than allowing the programmers to show what they can do – more suitable software might be produced. When designing IEM software it is all too easy to focus on the student (with respect to the user interface, etc.), and forget the teacher at another institution. The computer acts as tutor for a short while, but the cooperation of the educator is also vital to the success of an exercise, and they have their own goals for the subject, and thoughts on how it should be taught.

9.4.6 A Future for CAL3P

It should be remembered that the purpose of the various pieces of software created during this research was atypical. Where pages of information in a normal piece of IEM software would be standardised as far as possible, CAL3P deliberately explored several

different approaches to the provision of information. For this reason, CAL3P is not as effective as it might be; where any testing carried out during development would normally steer the development of a package towards the use of whichever media and presentation styles were found to be most popular and effective, CAL3P continued to offer a medley of techniques.

Despite being surprisingly effective and popular with students, in its current form CAL3P is far from ideal and would require further development work to realise its full potential. In particular, the removal of the digital video clips is recommended, since these were found to be ineffective, despite requiring a great deal of storage space. Also, it is suggested that the 'trick questions' in the built-in quiz (see Section 7.1.1) would need to be replaced – it would be pointless to require such questions to be answered if software trials were complete.

If resources allowed CAL3P to be modified more drastically, it should be changed to take advantage of the findings of this work, making greater use of the media which had produced the most effective learning, and reducing the emphasis of those which had proved less successful. This process of standardising on certain styles would have the advantage of simplifying the user interface, and could offer economies of scale during development. One aim in future programming should be to maximise the use of interactive elements, since these had demonstrated the most effective learning. Ideally, it should be possible to experiment with each process on the page where it is described; CAL3P currently shows animated machining sequences only when a student enters suitable instructions in their process plan and visits the simulation screen. Allowing the student to 'play' with each machining process separately would offer a shorter learning cycle which is less prone to confusion. The existing machining simulation would still be valuable, showing how complex parts can be made with a sequence of instructions.

In future development of CAL3P and perhaps IEM software as a whole, the degree of user interaction employed in each section should be considered in the same way in which each medium might be evaluated; with its own set of principles, technical implications, advantages and required skills in creation. Like other media, interactivity will not be used in isolation and it will not always be appropriate, but this work has shown how closer involvement on the part of the student can be positively encouraged, significantly improving the effectiveness of the other media used – a capability which must not be ignored.

10.0 Conclusions

This chapter completes the thesis. The project deliverables are identified, plus findings and major contributions to science, engineering and education.

10.1 Deliverables

- The CAL3P software itself is perhaps the most visible of the items delivered during the course of the research. It was demonstrated on many occasions, and used to teach over sixty students, with considerable success.
- The DICTA Information System was demonstrated extensively, and some of its novel features would later reappear in the MIDAS project software. (MIDAS was the successor to the DICTA project at the University of Salford.)
- In an effort to transform IEM development from a black art into more of a science, the essential principles governing the use of each medium were drawn together from a number of sources; see Sections 8.4.1 – 8.8.1. The advantages and disadvantages of each medium were discussed, with particular reference to the requirements of an academic IEM developer.
- Media were assessed by experimental means, identifying which contributed the greatest amount to a learning process. Time measurement carried out during software development then allowed the effectiveness of each approach to be contrasted with the cost involved.
- Six papers were presented at conferences during the period covered by this thesis; four on the subject of multimedia in education and two relating to the uses of IEM in a design situation. This was an important activity, disseminating the findings of the research to the community of educators who are making use of multimedia technology.

10.2 Findings

Observations have been made throughout the thesis, but in the interest of clarity, the findings of the research are emphasised in the three subsections that follow. The validity of the methodology which was employed is discussed, and the findings relating to the usefulness of interactivity and the various media are restated.

10.2.1 Methodology Employed

- Research into the use of IEM in engineering was achieved via two principal case studies; the first addressed the testability of printed circuit board designs, and the second concerned the education of novice process planners.
- For each case study, a new piece of IEM software was developed, demonstrating the technologies available and in fact pushing the boundaries of educational multimedia to some extent by including dynamically generated content.
- In order to prove that the CAL3P software was functioning satisfactorily, and to gather experimental data concerning the effectiveness of various media employed within the program, student attainment testing was carried out. The students' answers to a computer-based quiz were recorded, and subsequently provided valuable insights into how future IEM software might be improved. (The experimental process is described in greater detail in Chapter 7.)
- The computer-based quiz in CAL3P featured a novel method of rapidly identifying students with prior knowledge, as described from Section 7.1.1. This was important since it would be unreasonable to claim that learning had taken place simply because a student appeared to be knowledgeable at the end of the lesson. It also circumvented the ethical dilemma of splitting a group of students and teaching them in different ways for the purpose of establishing a control group.
- Performance among those identified as novices was compared, on a question-by-question basis, with the method(s) by which the training which could produce a correct answer had been supplied. In this manner, the value of several key components of IEM was investigated, to a hitherto unseen level of detail.

10.2.2 Interactivity

- The software which was produced featured a very high level of interactivity, for an IEM application; off the scale of Schwier and Misanchuk's [1993] 'Taxonomy of multimedia interaction' (see Section 4.6).
- It was clear that the motivation of learners was achieved, this accomplished by producing animations and screen displays in direct response to each student's input. Using CAL3P became a challenge; a game in which students were rewarded by seeing the product take shape, and penalised by seeing their work spoilt if they selected the wrong processes or parameters.

-
- In addition to their effectiveness, the interactive animations produced within CAL3P were found to require substantially less computer resources than the digital video which might otherwise have been employed. To a developer who anticipates that his audience might not be equipped with the latest computers, this difference could be critical.
 - Development effort for interactive elements was found to be very high; Section 8.9.6 reported the ratio of coding time to playback time for an interactive animation was in the region of 3600:1. It was argued, however, that the high level of flexibility brought about as a result mitigated this problem to a great extent. Animations were likely to be viewed many times in the course of an iterative learning process, and the software could subsequently be used to plan the manufacture of an entirely different part.
 - As a consequence of the tremendous success enjoyed with dynamically generated animations, it was proposed that interactivity be considered a medium in its own right. This is reasonable since the inclusion of interactive elements involves additional development work, requires additional storage space and has a measurable effect – acting in these respects like text, photographs or any other medium.
 - That interactivity would be so extraordinarily effective had been to some extent predicted by learning theory, which had demanded a cyclic process involving observation, theory building, experimentation and experience. That a computer-based learning environment should match the human learning process is only reasonable.

10.2.3 Positive and Negative Aspects of the Different Technologies' Impact on IEM

- Of the media available to the IEM software developer, text was found to be the least effective. Section 8.9.1 describes the relatively poor results which were obtained. Text was, however, by far the simplest medium to add to a piece of IEM software, in part because so much educational information already exists in that form and is readily transcribed.
- Graphics were more effective than text, and diagrams proved to be more effective than photographs. It was suggested that this was due to the simplicity of diagrams; photographs tend to include extra elements which can be a distraction.
- With regard to moving images, animated vector graphics were found to be more effective than the commonly used alternative of digital video. Again, this

may be due in part to the clarity of specially drawn media, but it should also be remembered that CAL3P's animations were highly interactive.

- Experimental results presented in Section 8.9.3 show that the students who had the opportunity to view digital video clips performed worse, over a sample of five questions where the correct answer might be seen in a video clip, than those who were forced to learn by other means. This was stunning; the presence of an extra source of information appeared to actually lower students' performance! The students who used the video clips were not working under a time limit, so there is no reason why they should not have performed at least as well as the others; superior performance had been expected.
- In addition to being more effective, it was clear that the synthetic media (diagrams, animations) made less demands on the computer; this may mean that a classroom equipped with slower machines can still be useful, or that a larger variety of information can be stored.
- Despite being identified as less effective, photographs and video were found to be somewhat easier to create than their diagrammatic and animated equivalents. Simply pointing a camera at something will generally be a lot quicker than drawing it, particularly in the era of affordable digital cameras. Of course, having the necessary equipment does not guarantee that high quality IEM software will result. (Sections 8.5.1 and 8.6.1 suggest some good working practices.)
- Despite the poor performance of video in the CAL3P trials, the medium should not be entirely discounted. As Section 8.6.4 describes, video was the only medium by which some details of machining processes could reasonably be shown. Whatever the subject, there will always be instances where it would be extremely difficult to demonstrate something by means other than video. Imagine trying to show how a horse gallops – it would be very difficult (and expensive) to create a convincing animation.
- No medium is universally applicable, and it is likely that over-reliance on a single medium will reduce its impact and lower the effectiveness of the software as a whole. The guidelines presented in Chapter 8 may aid developers of IEM software to make the best use of each.

10.3 Major Contributions to Science, Engineering and Education

- The research described in this thesis revealed several common assumptions about IEM, and placed them under scrutiny; that any good teacher with the

inclination could create a multimedia application, that it would certainly improve learning, and that it might offer a cost-effective means of instruction. To address these in turn:

- It was found that opinions such as those of Bhide and Nene [1997] (see Section 8.1.3), that anyone who can teach can create multimedia are not entirely correct. Any good teacher can generate some content for IEM, but without drawing skills, interface design skills or an appreciation of learning theory, the result is likely to be disappointing to all concerned. Section 3.6 contained warnings to this effect, from Turnbull [1974], Hudson [1984], Fisher [1994] and Bradshaw [1996].
- It would be heroic to suggest that IEM will always improve learning. In the cases investigated it was found that IEM can improve communication and education, but few other authors had undertaken any real form of attainment testing.
- As Section 3.7.1 discussed, multimedia education may be economically viable if the number of students who ultimately use the software is huge, as it was for American Airlines in the case described by Cillo [1991]. To the academic software developer, this is simply not the case; research centred upon CAL3P went some way towards establishing a model where the total amount of effort required to add a feature to an IEM package could be estimated, and the likely educational consequences predicted.
- If, despite the warnings given above, a package is to be created, the guidelines presented in this thesis will be valuable to the developer, identifying good practices and, with regard to engineering education in particular, the media and methods which were most effective.
- It must be concluded that the models of the learning process, originally developed by educational psychologists to describe conventional instruction, are valid in this new field. Facilitating a learning process of the type described by Kolb [1974] was not the simplest way to produce IEM instruction, but it proved to be technically possible. In terms of educational effectiveness, an iterative learning process was advantageous.
- The experimental method, using a computer-based quiz and special 'trick questions' (see Section 7.1.1) which identified learners who had prior knowledge of the subject represented a significant advance, allowing

attainment testing to be conducted without the need to split the learners into two or more groups.

- To educators and the suppliers of multimedia authoring software, CAL3P demonstrated a much higher level of interactivity than was normally seen, generating new content in direct response to a user's entries. This was shown to be a powerful way to facilitate learning, which other developers would be well advised to follow.
- The author hopes that this thesis has demonstrated a way forward to future IEM applications which are better targeted and more cost-effective. The people who choose to develop a piece of IEM software should choose their ground with care; CAL3P has demonstrated how the aim should not be to automate a human-based process, but to reengineer a learning process such that it takes advantage of the new opportunities offered by the computer, and avoids the shortcomings of the same.

Glossary of Terms

Animation	A method of showing moving images on the computer screen. In this thesis, the term animation has been used to refer in particular to the movement of elements of a diagram, achieved by programming. Where a moving image was displayed by playing back a digital video file, it is more appropriate to consider the medium a movie.
Application	A computer program; for example, CAL3P could be referred to as a training application.
ASCII	The American Standard Code for Information Interchange; a system of coding of letters, numbers and other symbols. ASCII provides a useful common format when moving information between programs or platforms.
Audio	See 'sound'.
Authoring	The process of creating a multimedia document, primarily referring to programming work but also including operations such as preparing content or managing the project. Authoring requires input from people with expertise in writing content, shooting and editing video, recording and editing audio, programming, graphic design, image processing and other areas.
AVI	'Audio Video Interleave' – a common format for the storage of digitised video clips.
Baud	A measure of the transmission rate of data, in bits per second.
Bit	A binary digit; the smallest unit of information.
Bitmap, bitmapped graphic	An image appearing on the computer screen, made up from a rectangular array of pixels.
Branching	Moving between pages or sections of information in a hypermedia environment, where multiple routes through the information are available.

Browsing	See 'navigation'.
Byte	A piece of data 7 or 8 bits long, perhaps used to store a single alphanumeric character. 1024 bytes make a kilobyte.
CAD	Computer Aided Design. Use of a computer to store drawings and design models, allowing a user to manipulate the data as required.
CAL	Computer Aided Learning. Other acronyms include CBT, CAT, CMI, CAI... these all refer to essentially the same thing; the use of digital computers in education.
CAL3P	'Computer Aided Learning of Process Planning Principles' – a piece of software created during this research. See Chapter 5.
Card	One screen of information in HyperCard, a multimedia authoring environment which organises information on a page-by-page basis, as opposed to using a time based approach.
CD-Rom	Compact Disk, Read-Only Memory. A common digital format for the distribution of multimedia information, since it combines large capacity (typically 660MB per disk) with random access. CDs are small, robust and can be read at a reasonably high speed.
Compression	The encryption of data to reduce the amount of storage space required. This is of particular importance in multimedia as large numbers of digitised photos and video sequences are often required. Common formats to reduce storage requirements for these media are JPEG and MPEG respectively.
Cursor	A small graphic which may be moved around the computer screen (typically by moving a mouse), allowing the user to make selections in a graphical interface.
DCG	Dynamic Content Generation – producing new media in response to the entries made by the user, as opposed to

	playing prerecorded media. CAL3P produces some of its text, graphics and animations in this way.
Dialog box	A pop-up graphic used to report something to a user, or ask for information. The rest of the screen is usually disabled while a dialog box is visible. Making a choice or entering a required piece of information will close (hide) the dialog box and allow the user to continue.
Digital	Digital information is stored in the form of a sequence of numerical values, as opposed to a continuously variable stream of analogue information. Digital storage and reproduction processes are generally free from degradation.
DIS	The 'DICTA Information system' – a piece of software created during this research. See Chapter 4.
Expert System	An interactive program which diagnoses problems based on the user's responses to questions, using a knowledge base built to represent the decision processes of a human expert in the field.
Gigabyte (GB)	A measure of computer storage; 1024 megabytes is a gigabyte. This is currently the largest commonly-used measure of storage space, typically only required to describe the capacity of hard disk drives.
Graphics	Illustrations appearing on the computer screen. In this thesis, 'graphics' generally refers to diagrams created on a computer, as distinct from digitised (scanned) photographs.
Hardware	The physical computer system being used, including peripheral devices. See 'MPC'.
Hotspot	An area of the computer screen which may be selected with the cursor, typically activating a script of facilitating navigation to a new section of information.
Hypermedia	"Hypertext and Multimedia". A document containing various media forms on the same subject, which can be browsed selectively in a non-linear manner.

Hypertext	The predecessor to hypermedia. An interactive document, but containing text only. Still in use in areas such as on-line documentation for software.
Icon	A small picture, typically illustrating an on-screen button. The icon should convey the purpose of the button.
IEM	Interactive, Educational Multimedia – CAL software which uses a mixture of media.
Interactivity	The degree of communication between user and computer, allowing choice of sections which might be of interest, requiring the completion of on-line tests, etc.
Interface	The system whereby user and computer communicate, comprising input devices such as the mouse and keyboard, and outputs such as suitably designed on-screen menus.
Internet	The largest worldwide electronic network, comprising thousands of smaller networks and millions of computer users. Applications include business, education and entertainment.
JPEG	Joint Photographic Experts Group. A popular compression format for use with still images.
Kilobyte (K)	A measure of computer storage; 1024 kilobytes is a megabyte (MB).
Media	Plural of Medium: the forms by which information might be conveyed.
Megabyte (MB)	A measure of computer storage requirements or capacity; a multimedia computer program will require a given amount of RAM and disk storage, probably given in megabytes. 1024 kilobytes makes a megabyte.
MIDI	‘Musical Instrument Digital Interface’ – a format which allows music to be stored or transmitted in a compact manner, and reproduced on equipment from various manufacturers. MIDI files have the advantage of being

much smaller than digital audio files, though they are not always suitable.

MPC and MPC2

Early standards for a multimedia platform based on IBM PC compatible computers. MPC2 requires at least a 25MHz 486SX processor, 4MB of RAM, a 1.44MB floppy disk drive, 160MB hard disk drive, a double speed CD-Rom drive, 16 bit sound board, 640 x 480 screen capable of displaying 16 bit colour, a full keyboard, a 2 button mouse, serial, parallel, MIDI and joystick ports, and Windows version 3.1.

MPEG

Motion Picture Experts Group. A format for the compression of movies, with a compression ratio of about 50:1.

Multimedia

The combination of several output forms within one document or application. Possible media include text, audio, video and graphics.

Navigation

Movement within multimedia software to reach other sections which are of interest. Navigation is generally achieved with hypertext links or on-screen buttons.

Pixel

Abbreviation of "Picture element". A single dot on the computer screen. The resolution of a piece of display hardware is measured as the number of rows and columns of pixels which it can display.

Platform

The combination of hardware and software on which multimedia software can be created or displayed. Examples include Macintosh, Multimedia PC and CD-I.

Program, Programme

A programme is an organised series of events. This conventional spelling is also used in this country to describe television entertainment. In the US, the spelling 'Program' is used exclusively, and this has come to be used in computing in both countries. Both spellings appear within this report; the author has used 'programme' in reference to a predefined stream of

	information, while 'program' refers to the code at work within an application.
QuickTime	A popular compression format, primarily for the storage of video clips.
RAM	Random Access Memory. The part of a computer where information may be temporarily stored for manipulation.
Script	A part of a multimedia program, often assigned to a specific element such as a button.
Sound	Speech, sound effects or music, stored in a digital form and reproduced as part of a multimedia programme (see also 'MIDI'). This medium may also be referred to as Audio.
Storyboarding	A development activity, planning the architecture of a multimedia programme (see Appendix 1). In the storyboard, screen layouts are sketched and annotated to show how elements of the user interface should function.
Text	Alphanumeric information, such as could be entered via a keyboard (text in this context includes numbers). This medium has advantages of efficiency in storage and (sometimes) clarity. Additionally, it can typically be read by outdated equipment, when other media may be impossible to display. Text is poor at conveying motion and representing time-dependent or spatial relationships.
Vector graphic	An image made up from lines, polygons, arcs etc., typically more simplistic than a bitmapped graphic, but this can be advantageous, offering clarity and low storage requirements.
Video	The display of moving visual images. In the context of multimedia, these are digitised film clips, stored in a format suitable for playback as part of a multimedia programme, such as AVI, MPEG or QuickTime.

Viewer

The software used for navigating through a multimedia document. This may well be a cut-down version of the authoring software, called a Player.

References

- Abelson H and Abelson A (1992) "Logo for the Macintosh, An Introduction Through Object Logo", Paradigm Software Inc, Cambridge MA
- Adcock CJ (1990) "Fundamentals of Psychology", Penguin Books
- Amore RC (1998) "Designing Training Programs for Factory Workers", proceedings of On-line Europe '98
- Atkinson RL, Atkinson RC, Smith EE and Bem DJ (1993) "Introduction to Psychology", 11th edition, Harcourt Brace & Company
- Ausubel DP (1968) "Educational Psychology: A Cognitive View", Holt, Rinehart and Winston, New York
- Avner RA (1988) "Is there an ideal size for courseware production teams?" Proc. 30th Association for the Development of Computer Based Instructional Systems (ADCIS), Philadelphia, 1988
- Baddeley A (1990) "Human Memory, Theory and Practice", Erlbaum, London
- Bahrck HP (1984) "Semantic memory content in permastore: Fifty years of memory for Spanish learned in school", Journal of Experimental Psychology, Vol. 113 pp1-29
- Beevers CE, Cherry BSG, Foster MG and McGuire GR (1991) "Software Tools for Computer Aided Learning in Mathematics", Avebury Technical
- Bhide SR and Nene AS (1997) "Some Experiences in Developing CAL Packages for Engineering Education", Proceedings of the 3rd International Conference on Computer Based Learning in Science (CBLiS '97), DeMontfort University, Leicester, 4th - 8th July 1997
- Boehm BW (1981) "Software Engineering Economics", Prentice-Hall
- Borenstein NS (1991) "Programming as if People Mattered: Friendly Programs, Software Engineering, and other Noble Delusions", Princeton University Press
- Bower GH (1977) "A multicomponent theory of the memory trace", in "Human Memory: Basic Processes (Bower GH, editor), Academic Press, Boston
- Bradshaw M (1996) "The design of computer-based learning programs: a solution to the problem of reproducing teaching-learning acts through the computer interface", Proceedings of the ALT Conference, Strathclyde Glasgow and Caledonian Universities, 16th - 18th September 1996
- Bradshaw M (1997) <http://www.netcentral.co.uk/pmb> (web pages)
- Bullock M (1987) "Design SMT Boards for In-Circuit Testability", Paper 25.1 in Proceedings of the 1987 International TEST Conference, IEEE

-
- [Burger J](#) (1993) "The Desktop Multimedia Bible", Addison-Wesley, New York
- [Bush V](#) (1945) "As We May Think" Atlantic Monthly, July
- [Buzan T](#) (1993) "The Mind Map Book", BBC Books
- [Buzan T](#) (1974) "Use Your Head", BBC Books
- [Cann AJ](#) (1997) "Online Interactive Computer-Assisted Learning", Proceedings of the 3rd International Conference on Computer Based Learning in Science (CBLiS '97), DeMontfort University, Leicester, 4th – 8th July 1997
- [Carmichael L](#), [Hogan HF](#) and [Walter AA](#) (1932) "An experimental study of the effect of language on the reproduction of visually perceived form", Journal of Experimental Psychology, Vol. 15, pp73–86
- [Chang TC](#) and [Wysk RA](#) (1985) "An introduction to automated process planning systems", Prentice-Hall, New Jersey
- [Cillo G](#) (1991 January) "Apple: 'Multimedia Integration' is what we do: More to come" Computer Picture, pp12–16
- [Clarke D](#), [Turban E](#) and [Wang P](#) (1994) "Integrating Expert Systems and Multimedia for Improved Troubleshooting of the City of Los Angeles's Computer Hardware", Expert Systems with Applications, Vol. 7, No. 3, pp441–449.
- [Cohen VB](#) (1985) "A Reexamination of Feedback in Computer-Based Instruction: Implications for Instructional Design", Educational Technology, January, pp33–37
- [Collins SMH](#), editor (1996), "Dictionary of Information Technology", 2nd edition, Peter Collins Publishing Ltd.
- [Cotton B](#) and [Oliver R](#) (1997) "Understanding Hypermedia", 2nd edition, Phaidon Press, London
- [Crampes M](#) (1991) "The Operational Nature of Methodman I: A training tool in project management", EC COMETT project report 91/1/5081/Cb
- [Cross N](#), [Durling D](#) and [Johnson J](#) (1996) "CAI with Style", Proceedings of SEED '96, Sharing Experiences in Engineering Design, University of Bristol 26-27th June 1996
- [Culverhouse PF](#) (1993) "Four design routes in electronics engineering product development", Journal of Design and Manufacturing Vol. 3, pp147–158
- [Currie MA](#), [Ekere NN](#), [Mannan SH](#) and [Ismail I](#) (1994) "Solder paste characteristics and their effect on fine pitch printing", Proceedings of the 4th International Conference on Factory 2000 – Advanced Factory Automation, University of York, 1994
- [Davies ML](#) and [Crowther DEA](#) (1995) "The benefits of using multimedia in higher education: myths and realities" in Active Learning (No. 3), CTI Support Services, pp3–6

-
- deCaluwe N, Farr R and Barber M (2000) "Improved Circuit Board Design through the use of the MIDAS Critiquer and Multimedia Environment", Proceedings of the 3rd International Symposium on Tools and Methods of Competitive Engineering (TMCE 2000), Delft University of Technology, Holland, 18th – 21st April 2000
- DeGarmo EP, Black JT and Kohser RA (1990) "Materials and Processes in Manufacturing", 7th edition, Macmillan, New York
- Derville LMT (1966) "The use of Psychology in Teaching", Longman, London
- Earnshaw RA (1993) "The exploitation of new media for text, graphics, images and sound", Advisory Group on Computer Graphics, Loughborough
- Ebbinghaus H (1913) "Memory", New York (Translation from the German "Über das Gedächtnis", Dunker, Leipzig, 1885)
- Estes WK (1972) "An associative basis for coding and organisation in memory" in "Coding processes in Human Memory" (Melton AW and Martin E, editors)
- Falk DR and Carlson HL (1995) "Multimedia in Higher Education, A practical guide to new tools for interactive teaching and learning", Learned Information Inc.
- Farr R and Lawlor-Wright T (1996) "A Computer Aided Learning System for Process Planning", Proceedings of SEED '96, University of Bristol 26-27th June 1996
- Farr R and Lawlor-Wright T (1997a) "An Interactive Computer Aided Learning System for Manufacturing Process Planning", Proceedings of the 3rd International Conference on Computer Based Learning in Science (CBLiS '97), DeMontfort University, Leicester, 4th – 8th July 1997
- Farr R and Lawlor-Wright T (1997b) "Use of Multimedia to Support Teaching of Manufacturing Process Planning", Advances in Manufacturing Technology XI, Proceedings of the Thirteenth National Conference on Manufacturing Research, Glasgow Caledonian University, 9th – 11th September 1997
- Farr R and Lawlor-Wright T (1997c) "A Computer-Aided Learning System to Promote Design for Manufacture", Proceedings of the International Conference on Engineering Design (ICED), Tampere, Finland, 1997
- Farr R and Lawlor-Wright T (1998) "Computer Aided Learning of Process Planning using a 'Virtual Machine Tool'", Proceedings of the On-line Learning, Europe conference (on CD-Rom), Cheltenham and Gloucester College of Higher Education, July 1998
- Fischer G, Nakakoji K, Ostwald J, Stahl G and Sumner T (1993) "Embedding Critics in Design Environments", The Knowledge Engineering Review, Vol. 8, Part 4, pp285–307
- Fisher S (1994) "Multimedia Authoring, Building and Developing Documents", Academic Press Inc., Boston

-
- Fleishman EA and Parker JF (1962) "Factors in the retention and relearning of perceptual motor skill", *Journal of Experimental Psychology* Vol. 64, pp215–226
- Gagné RM (1975) "Essentials of Learning for Instruction", Holt Rinehart and Winston, New York
- Gallager C, Lawlor-Wright T and Molloy O (1994) "Issues in the Development of Knowledge-based Systems for Design for Assembly and Testability of Printed Circuit Boards", *Proceedings of the 2nd International Conference on Concurrent Engineering and Electronic Design Automation*, Bournemouth, UK, 9–11th April 1994, pp81–87
- Gallager C and Lawlor-Wright T (1995) "A Review of PCB Design for In-Circuit Testability Guidelines and Systems", *Journal of Electronics Manufacture*, Vol. 5 No. 3
- Gallagher C (1996) "The integrated design of printed circuit boards and their in-circuit test fixtures", Ph.D. Thesis, University of Salford
- Goble C (1994) "Multimedia Information Systems: the confluence of technologies and disciplines", *Information and Software Technology*, Vol. 36 No. 4, pp195–196.
- Godden D and Baddeley AD (1975) "Context-dependent memory in two natural environments: On land and under water" *British Journal of Psychology*, Vol. 66 pp325–331
- Golas KC (1993) "Estimating time to develop interactive courseware in the 1990s" *Proceedings of the Interservices/Industry Training and Education Conference*, 1993
- Haber RN (1970) "How We Remember What We See", *Scientific American*, Vol. 105, May 1970
- Haddon K, Smith C, Brattan D and Smith T (1995) "Can learning via multimedia benefit weaker students?" in *Active Learning* (No. 3), CTI Support Services, December 1995, pp22–27
- Hall M and Robinson D (1995) "The Multimedia Brain", *Active Learning* (No. 3), CTI Support Services, December 1995, pp36–41
- Handy CB (1989) "The age of Unreason", Business Books Ltd.
- Hargreaves R (unpublished) On-line documentation for ICL design staff, ICL Corporate Systems, West Gorton
- Heath S (1996) "Multimedia & Communications Teachnology", Focal Press
- Hilgard ER and Bower GH (1975) "Theories of Learning", 3rd edition, Prentice-Hall
- Holland JG (1962) "Characteristics of some recent studies of instructional methods" in "Proceedings of the Conference on Application of Digital Computers to Automated Instruction" (Coulson JE, editor), System Development Corp

-
- Honigsbaum M (1992 July) "Set yourself new challenges to continue learning and stay happy", GQ Magazine
- Horton WK (1994) "Designing and writing online documentation: Hypermedia for self-supporting products" 2nd edition, Wiley, New York,
- Hudson K (1984) "Introducing CAL: a practical guide to writing computer assisted learning programs", Chapman and Hall, London
- Hunter IML (1957) "Memory: Facts and Fallacies", Penguin Books
- Illingworth V, Glaser EL and Pyle IC (1990) "Dictionary of Computing (Oxford Reference)", 3rd edition, Oxford University Press
- Inter-University Committee on Computing (1991) "Report of the Working Party on the Provision of Computing Facilities for Teaching", September 1991
- Jasperse HB (1994) "A Macro Process Planning System for Machining Operations"
- Kant I (1781) "Kritik der reinen Vernunft", Leipzig. Translation "Critique of Pure Reason", London, 1887
- Katz YJ and Offir B (1994) "Computer Games as Motivators for Successful Computer End-Use", IFIP Transactions A - Computer Science and Technology, Vol 58, pp 81–87
- Kolb DA (1974) "Organisational Psychology: An Experimental Approach", Prentice Hall
- Kong H (1994) "An Intelligent, Multimedia-Supported Instructional System, Expert Systems with Applications, Vol. 7, No.3, pp451–465
- Leonard J (1994) "Interacting: Multimedia and Health", BPCC Wheatons Ltd, Exeter
- Levin JR and Kaplan SA (1972) "Imaginal Facilitation of Paired-Associate Learning: a Limited Generalization?", Journal of Educational Psychology, Vol. 63 No. 5, pp429–432
- Linton M (1975) "Memory for real-world events" in "Explorations in cognition" (Norman DA and Rumelhart DE, editors), Freeman, San Francisco
- Luther AC (1994) "Authoring Interactive Multimedia", Boston Academic Press Professional, Boston
- Marshall IM, Samson WB and Dugard PI (1994) "Predicting the development effort of multimedia courseware", Information and Software Technology Vol. 36 No. 5, pp251–258
- Marshall IM, Samson WB, Dugard PI and Lund GR (1995) "The Mythical Courseware Development to Delivery Time Ratio", Computers Education Vol. 25 No. 3, pp113–122
- McKenna SP and Glendon AI (1985) "Occupational first aid training: Decay in cardiopulmonary resuscitation (CPR) skills" Journal of Occupational Psychology, Vol. 58 (1985) pp109-117

-
- Murdock BB** (1965) "The retention of individual items", *Journal of Experimental Psychology*, Vol. 62, pp618-625
- Myers IB** and **McCauley MH** (1985) "A guide to the development and use of the Myers-Briggs Type Indicator", Consulting Psychologists Press, Palo Alto
- Nelson D**, editor (1983) "Report of a Working Party on Computer Facilities for Teaching in Universities", Computer Board for Universities and Research Councils
- Nicholls BS** (1997) "Development and Integration of Highly Interactive Computer-Assisted-Learning in Chemistry – The 'ChemiCAL' Project", Proceedings of the 3rd International Conference on Computer Based Learning in Science (CBLiS '97), De Montfort University, Leicester, 4th – 8th July 1997
- Nicholson R** and **Simpson A** (1992) "Using HyperCard to create a flexible learning package for statistics – costs, benefits and effectiveness", *Lecture notes in Computer Science*, Vol. 602, pp516–528.
- Nickerson RS** (1965) "Short-term memory for complex meaningful visual configurations: A demonstration of capacity", *Canadian Journal of Psychology*, Vol. 19, pp 155-160
- Novak G** and **Descharnais R** (1998) "Virtual Courseware for Science Education: Virtual Flylab and Virtual Earthquake", in "Selected Papers from the 9th International Conference on College Teaching and Learning", (Chambers JA, editor), Florida Community College, Jacksonville
- Odor JP** (1992) "Hypermedia: choices, creations, care and maintenance", Learning Technologies Unit, University of Edinburgh Employment Department, Sheffield: Moorfoot
- Parkinson B**, **Hudson P**, **Senior R**, **Barker R** and **Short C** (1996) "Teaching Concurrent Engineering Design Utilising a Virtual Environment", Proceedings of the SEED '96 (Sharing Experiences in Engineering Design) conference, University of Bristol, 26th – 27th June 1996
- Parkinson B**, **Hudson P**, **Senior R**, **Barker R** and **Short C** (1997) "The Use of Intelligent Agents to Aid Design to Manufacture", *Advances in Manufacturing Technology XI*, Proceedings of the Thirteenth National Conference on Manufacturing Research, Glasgow Caledonian University, 9th – 11th September 1997
- Parkinson B** (1996) "IDeR Design Builder" (software produced on a TLTP funded project), University of Hertfordshire
- Pattis RE** (1981) "Karel the Robot: a gentle introduction to the art of programming", Wiley, New York
- Peterson LR** and **Peterson MJ** (1959), "Short-term retention of individual verbal items", *Journal of Experimental Psychology*, Vol. 58, pp193–198

-
- Petty DJ (1983) "The Optimisation of Cutting Conditions in Multi-pass Turning", M.Sc. Thesis, UMIST
- Powell J and Newland P (1994) "Informing multimedia: a sensitive interface to data for construction design professionals", *Design Studies*, Vol. 15 No. 3, pp285–316.
- Pressey SL (1960) "Autoinstruction: perspectives, problems, potentials", in "Theories of Learning and Instruction" (Hilgard ER, editor)
- Printed Circuit Interconnect Federation (1996) "The Enlightenment of Albert", Logical Connections Ltd.
- Printed Circuit Interconnect Federation (1997) "The Enlightenment of Albert" [CD-Rom], Logical Connections Ltd.
- Quinn CN (1994) "Designing Educational Computer Games", *IFIP Transactions A - Computer Science and Technology Vol 59*, pp 45–57
- Robinson P (1990, February) "The Four Multimedia Gospels", *Byte Magazine*
- Rushby NJ (1983) "Computer-based Learning: State of the Art Report", Pergamon
- Schneiderman B (1980) "Software psychology", Winthrop Publishers Inc., Cambridge MA
- Schneiderman B (1987) "User Interface Design for the Hypertext Electronic Encyclopedia" in "Hypertext 87 Papers", University of North Carolina, pp 189–194
- Schwier RA and Misanchuk ER (1993) "Interactive Multimedia Instruction", *Educational Technology*, Englewood Cliffs, New Jersey
- Seale JK (1997) "Developing CBL Packages: A Reflective Practice Approach in Science Education", *Proceedings of the 3rd International Conference on Computer Based Learning in Science (CBLiS '97)*, DeMontfort University, Leicester, 4th – 8th July 1997
- Shim JP (1992) "Living up to the Hype" OR/MS today, February 1992, Lionheart Publishing, Atlanta
- Sipior J and Garrity E (1992) "Merging Expert Systems with Multimedia Technology", *Data Base*, Winter 1992, pp45–49
- Taylor FW (1907) "On the Art of Cutting Metals", *Transactions of the ASME*, Vol. 27
- Terman LM and Merrill MA (1937) "Measuring Intelligence", Harrap, London
- Tombaugh J, Lickorish A and Wright P (1983) "Multi-window displays for readers of lengthy texts", *International Journal of Man-machine Studies*, Vol. 26 No. 5 pp 597–615
- Trabasso T and Bower GH (1975) "Attention in Learning: Theory and Research", Wiley, New York
- Trainor AJ and Dore AM (1994) "Interfacing with VR-enabled Robot Modelling and Simulation Environments" *Proceedings of EURISCON '94, European Robotics and Intelligent Systems Conference, Malaga, 1994, Vol. 1*, pp210–220

-
- [Trainor AJ](#), [Sharp JM](#) and [Dore AM](#) "The Inclusion of Personality Trait Based Adaptive Interfaces into Computer Based Learning Environments", in Proceedings of the 31st International MATADOR conference, April 1994, UMIST, UK
- [Travis LC](#) and [Jones PD](#) (1997) "From classroom to the manufacturing workplace using Multimedia for tuition", Advances in Manufacturing Technology XI, Proceedings of the Thirteenth National Conference on Manufacturing Research, 9th – 11th September 1997, Glasow Caledonian University, UK
- [Turino J](#) (1991) "Design for test pays cost, time dividends", Computer Design, 14th January 1991
- [Turino J](#) (1990) "Design to Test: A definitive guide for electronic design, manufacture and service", 2nd edition, Van Nostrand Reinhold, New York
- [Turnbull JJ](#) (1974) "The Computer in the Learning Environment", NCC Publications
- [University Grants Committee and Computer Board for Universities and Research Councils](#) "Report of a Joint Working Party on Teaching Computing in Universities", 1970
- [Vanegas JA](#) and [Baker NC](#) (1994) "Multimedia in Civil Engineering" Civil Engineering, Vol. 64 No. 5, pp71–73
- [Vaughan T](#) (1994) "Multimedia: Making it Work", 2nd edition, Osborne McGraw Hill
- [Vernon MD](#) (1971) "The Psychology of Perception", 2nd edition, Penguin
- [Walker PMB](#), general editor (1995) "Larousse Dictionary of Science and Technology", Larousse, New York
- [Wheldall K](#) (1983) "A Positive Approach to Classroom Discipline" in "Psychological Aspects of Learning and Teaching", (Wheldall K and Riding RJ, editors)' Croon Helm, London, pp10–39
- [Wu B](#) (1992) "Manufacturing Systems Design and Analysis", Chapman and Hall
- [Yolles MI](#) (1997) "Learning Style and Strategy, and the Theory of Viable Learning", Proceedings of the 3rd International Conference on Computer Based Learning in Science (CBLiS '97), DeMontfort University, Leicester, 4th – 8th July 1997
- [Zhang H](#) and [Alting L](#) (1994) "Computerised Manufacturing Process Planning Systems", Chapman and Hall, London

Appendix 1: Multimedia Software Architecture

In discussing a piece of IEM software's level of interactivity, it is sometimes necessary to describe multimedia programmes in terms of their architecture; the way in which navigation between sections of information may be achieved. To some degree, the level of interactivity desired will govern the software architecture selected, but it will not necessarily be the sole influence. This appendix describes several generic architecture formats.

Page-based and time-based paradigms

At the most fundamental level, IEM software may be organised in two ways; with a page-based or time-based structure.

A page-based architecture organises information into a series of separate screens. The user experiences the content of a page (reads text, listens to narration, etc.) and then advances to a new page for further information, typically by pressing a key or making a selection with a mouse.

A time-based system does not organise information into separate pages. Instead, it presents a single page. Multimedia elements are exhibited according to a timer; a text box might appear after at the start of the programme, and then fade into invisibility after 45 seconds, while a soundtrack starts playing after 20 seconds, and continues until the end of the file is reached... if writing a page-based multimedia application can be considered to be like writing a book, creating time-based one is like directing a movie. Multimedia software which uses a timeline tends to be more attractive, but also more prescriptive, allowing the user less freedom to browse. This is ideal for advertising, but less suitable for most Computer-Aided Learning applications.

The sections which follow describe the options available when structuring a page-based multimedia application.

Linear Architecture

Perhaps the most simple architecture for a piece of IEM software is the 'electronic book' paradigm. The simplest implementation might resemble that shown in Figure 1. Pages of information are accessed sequentially, with the user's control limited to advancing through the pages.

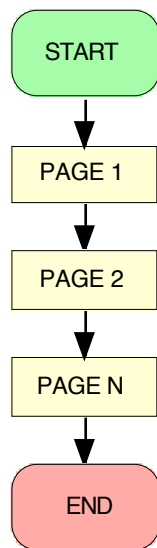


Figure 1: Sequential multimedia programme

This format can be achieved with very simple ‘authoring’ software such as Microsoft PowerPoint. Though the interactivity of the learning process is sorely limited, a full range of media types (text, graphics, movies, sound) may be employed in this kind of programme. Even if movie clips and audio are not employed, a programme with text and colour diagrams has certain advantages over a paper-based format; once created, it will be cheaper to duplicate and supply than an equivalent printed document, and it could be made available to a very large numbers of users if necessary.

Figure 1 suggests unidirectional travel through the document, but most programmes would feature some means for the reader to go back a page, in case they accidentally advanced the display before they had finished reading, or subsequently wanted to return to some earlier information and recap.

Branching Architecture

This most basic approach does not make significant use of the processing capabilities of the computer, and offering little that could not be achieved with an instructional video. An architecture which allows access to various sections which might be of interest is an improvement, as shown in Figure 2:

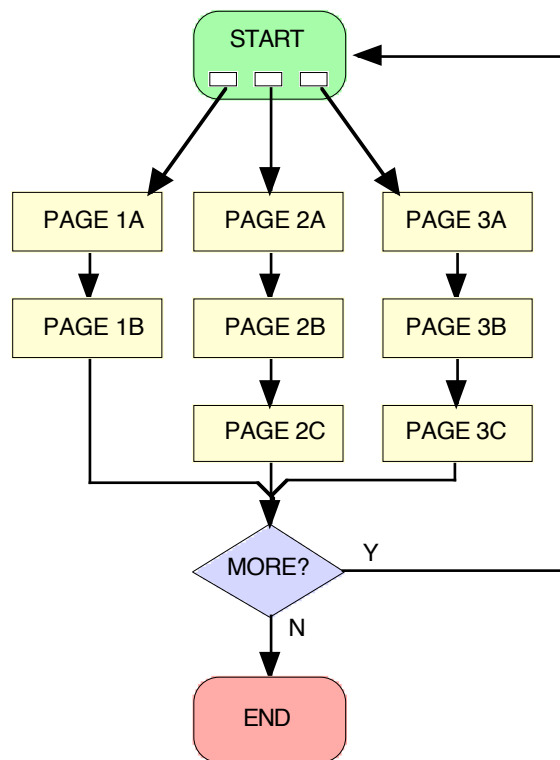


Figure 2: Sequential multimedia with choice of subject matter

The Inclusion of a Quiz

A quiz might be employed to determine whether the user should be allowed to advance to a new section, or if they have to reread the current one because they have not understood sufficiently. A model for software architecture of this kind can be seen in Figure 3.

An alternative use of a quiz is to produce a scored list of answers from which the user can identify weaknesses and draw their own conclusions. Some multimedia programmes have been built to record performance figures for students and using them to alert the staff members when a training need is identified [Beever et al., 1991].

Although it is a good idea to give the user feedback, an on-line quiz will usually be a poor substitute to human contact. Unless very carefully designed, a quiz will only assess short-term retention of facts rather than depth of understanding, and the abstract numerical score received can be misleading. Section 3.3.1 discussed the limitations of quizzes in CAL.

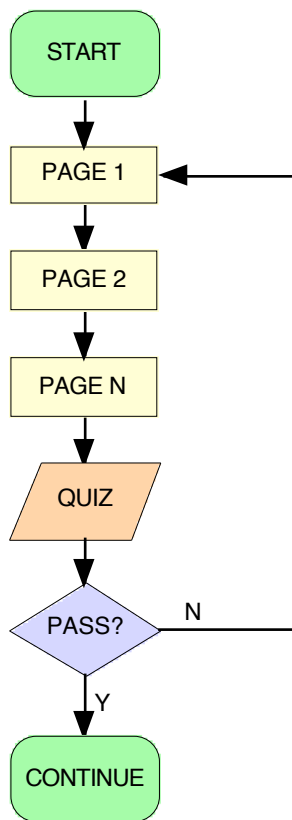


Figure 3: Linear software with a quiz function

Hypermedia

Hypertext (and, with multimedia files, Hypermedia) is meant to increase the quality of the learning experience because it allows the user to move about within the information presented, selecting areas of interest. With current technology, the usual method is to use a mouse controlled cursor to select one of a number of 'hot' areas on the computer screen, which are in effect bookmarks to other parts of the 'electronic book'. With navigable documentation, each learner is given opportunities to explore tangential information, allowing access to a variety of subjects that it would not be possible to cover in a normal lecture. Since the subjects chosen are often of personal interest, the learner's level of attention is increased. A schematic of the hypertext document structure is shown in Figure 4.

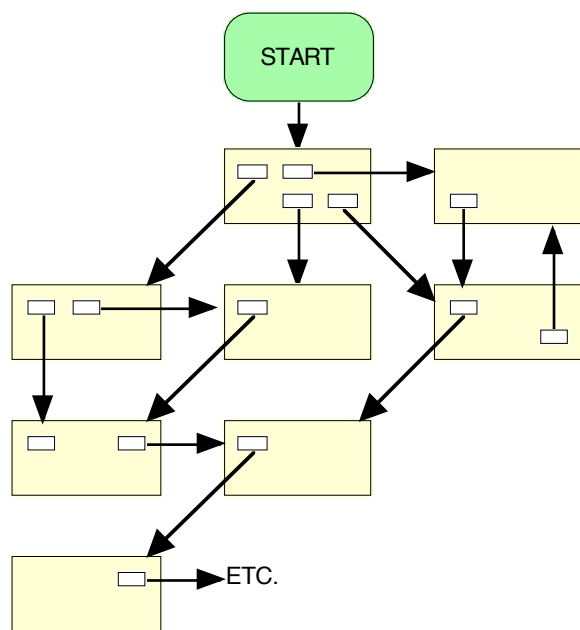


Figure 4: Pages of hypertext or hypermedia information

It must be borne in mind that the user already had this power to be selective when reading a book – to skim through something which they are confident that they already understand, to refer to an index and turn to a page or chapter of particular interest, and ultimately the power to put down the book and do something else. Hypermedia linking is a very useful tool on the way to effective computer-based education, but the concept of empowering the reader to choose the sections they want to read is nothing new in itself, since this can be done with any book that features an index.

Hybrid Architectures

Multimedia authoring packages are nothing if not flexible, and most are not restricted to only the architectures described so far. In addition, an application might be programmed to allow several different methods of navigation, or might make use of different styles in different modules. For instance, CAL3P's quiz is entirely linear, because the user is required to answer the questions in sequence, while in the 'Guide to Process Planning' part of the software, the user is allowed to navigate between sections which might be of interest.

Figure 5 shows how a largely sequential program could feature occasional hyperlinks, perhaps allowing the reader to move ahead if they found the current information too simple, while also offering the chance to recap when an unfamiliar concept is encountered. Both CAL3P and the DIS included occasional hyperlinks for this purpose, though experiments undertaken suggested that the educational value of

information accessed via hyperlinks was questionable because there was no guarantee that all information would be visited (see Section 7.9.7).

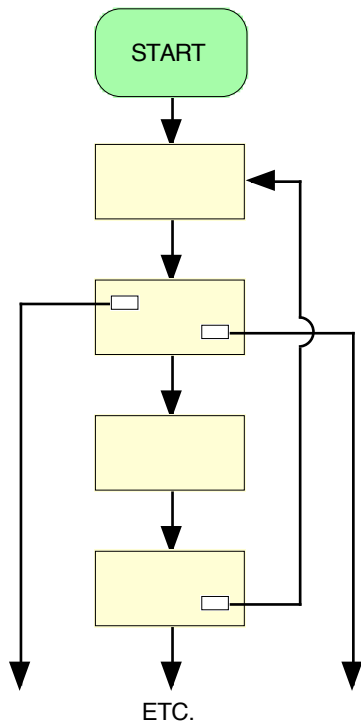


Figure 5: Sequential multimedia programme with occasional hyperlinks

Appendix 2: Multimedia Authoring Software Evaluation

During this work, a number of commercial authoring packages were considered for use. This appendix details the products reviewed. The reader should bear in mind that the range of authoring packages described represents those available at the time of software selection, in 1995. The review process and the criteria employed were described in detail in [Farr, 1996].

Because the project aims required the testing of a wide range of different media and presentation styles, it was necessary to select an authoring environment which allowed such content to be created and presented. Early prototypes of CAL3P and the DIS had shown that a highly interactive end product should be pursued, requiring a powerful scripting language, yet one which was reasonably easy to master. Naturally, since development effort had been identified as a limiting factor, the authoring package selected had to allow a reasonable rate of progress.

There were many pieces of authoring software which had a limited multimedia capability, including a number of slideshow creation packages which allowed the use of sound and film clips. Though on-line documentation produced with these might qualify as multimedia in the broadest sense, interactivity was too limited. Some of these programs allowed navigation between subsections of a presentation, but there was no capability to perform calculations, read and write files, or link to other programs. It was thus clear that these would not be an adequate environment for the creation of an IEM application. Packages rejected at this stage included PowerPoint, Super Show and Tell, Cinemania, Action and Dramatix, as described:

PowerPoint (Microsoft)

This was found to be easy to use, and good results were possible without great artistic flair due to the way diagrams were built up from layers of simple polygons which could be altered at any time. PowerPoint was used to create a slideshow of rules relating to the DICTA project. This was demonstrated and disseminated widely, but ultimately superseded by a system which could be integrated with CAD-based design advice, as described in Chapter 4. (Appendix 3 shows a selection of screens from the PowerPoint slideshow.)

Super Show and Tell (Ask-Me Multimedia)

This software was found to be easy to use, but very limited in function. All authoring was done within a single window, including features which allowed entities in the presentation to change over time, for instance changing position, size or colour to

achieve animation. Any object could be defined as a button and linked to another section of the presentation, or a separate presentation. Super Show and Tell's buttons also allowed sounds to be played, and movies to be shown, but there was no way to achieve more complex interaction. Work with this software could produce nothing more complex than a basic, informative document.

Cinematic (Vividus)

This program used a time-based rather than page-based authoring style, but the lack of any scripting facility meant it had to be considered with the 'slideshow' software rather than the true authoring environments, such as Macromedia Director, which share its time-based paradigm. One interesting feature was that Cinematic could import entire Microsoft PowerPoint or Aldus Persuasion files, converting each slide as appropriate. Given the wide variety of multimedia formats, this capability was convenient. Such imported files could have animation added with ease.

Action (Macromedia)

Another time-based authoring package, Action seemed well-suited to the creation of attention-grabbing visual media used to accompany a verbal presentation. It was found to include a variety of built-in special effects; objects could swoop around the screen, fade in and out, or sparkle to attract attention. More complex authoring, allowing even simple interaction such as branching between pages of interest could not be achieved with Action. Although available for Windows and Macintosh, the files created did not have cross-platform compatibility, which might have been a useful feature.

Dramatix (Zuken-Redac)

Dramatix was developed in-house by Zuken-Redac, an industrial partner on the DICTA project. It was not intended to be a commercial piece of authoring software, but was made available for use. Dramatix had a simple method of scripting whereby graphics files were named and any associated text specified, all in an ASCII file called 'dramatix.txt'. When the program was run, the images and text would appear on the screen, formatted as required.

Dramatix was a useful dissemination tool for Zuken-Redac; the program and sufficient files for a presentation that lasted five minutes or more could be held on one floppy disk (though accessing the files from floppy was slow). Unfortunately, no other media were supported and there was no way to interact with the presentation other than by clicking on a pair of buttons to go forward or backward through the sequence of screens.

...

Having discounted the products which were best suited only to the creation of simple slideshows, the multimedia authoring tools remaining to be considered were HyperCard, HyperStudio, Guide, Authorware, Director, IconAuthor, Multimedia ToolBook and HiPWorks.

HyperCard (Apple / Claris Corporation)

HyperCard was never intended to be a multimedia authoring tool – it is an environment best suited to database construction, yet it can be used for multimedia work because each card (record within the database) can include multimedia elements. Originally appearing in 1987, HyperCard was supplied free with Macintosh computers until 1991 (thereafter, Claris marketed an updated version and only a ‘player’ was given away free), HyperCard allowed a wide variety of applications to be created, all with the characteristic Apple graphic user interface. The user base was extensive, though Macintosh computers were found to be rare in UK education.

HyperCard did not support the creation or display of colour graphics fully. Good results could be achieved through the use of a plug-in called ‘Color Tools’, but including colour graphics such as photographs certainly complicated development. (HyperCard scripts could control full colour QuickTime movies, an industry standard.)

HyperCard was found to have an excellent scripting language, allowing complex calculations to be made and permitting direct control over presentations as desired. Section 5.3 described how scripted commands in the CAL3P prototype achieved complex, flexible animations. Since scripts were interpreted in runtime rather than executed in a compiled form (the same is true for almost all multimedia authoring languages), speed was a problem on older computers.

HyperStudio (Roger Wagner Publishing)

This package was considered superior to HyperCard in that it allowed full colour graphics to be displayed as standard. It also altered the screen display to suit the hardware available, which seemed an excellent way to reach a wide audience. Sound was better supported (HyperCard required that sounds be converted to a non-standard format), and video was presented in a slightly more polished way than within HyperCard. In effect, this more specialised software is better at multimedia. Development was fairly simple, allowing impressive multimedia presentations to be made without a

great deal of complex programming. Branching was possible, as were some other interactions with the user.

Unfortunately the scripting language, HyperLogo, would not allow the complex calculations which were required in this research. It was anticipated that building up a highly interactive program from the limited set of commands available would be arduous. It is unlikely that dynamic content generation of the kind exhibited by CAL3P could have been achieved.

Guide (Owl International, Inc.)

Although it was quite old even at the start of this research, Guide was evaluated because the software was already available locally and had been used for project work in the past. It was found that the software could achieve a moderate level of interactivity, and some elegant graphics and effects were seen, but it was disappointing in terms of meeting the criteria for 'multimedia', with no provision for animation. The power of Guide's scripting language, Logiix, was limited, concerned almost entirely with commands to control navigation through a presentation.

Guide would have allowed the creation of a good hypermedia document, but a real IEM application could not be created. No new version of the software had been released for some years, and Guide was no longer mentioned in current literature, so it seemed unlikely that the package would be improved.

Authorware Professional (Macromedia)

Some experimental programming was carried out using Authorware Star, a freely available trial release of the current version (then 2.2). This had all the features of the full program, except that the programmer was limited to a maximum of 500 nodes (each node is an element in the visual programming technique called 'Object Authoring'). This limitation was insignificant for evaluation purposes.

Object Authoring was found to be less easy than was claimed. This is a graphical programming technique where a flow chart of nodes is created by dragging them into place with the mouse. Creation of simple programmes was quite fast, but representing more complex operations such as a loop with several 'if... then' type statements was cumbersome, requiring large numbers of nodes. It was clear that the creation of a highly interactive application within this environment would be a lengthy process. Despite the claim of script-free authoring, some conventional scripting was still required in Authorware, within special calculation nodes which were used to perform mathematical functions. A typical operation handled by a calculation node might be to calculate marks after a test has been completed.

Authorware was an expensive option, despite the fact that the built-in editing facilities were of very poor quality, and the version available at the time of review did not include some facilities which were standard in other products, such as support for MIDI audio.

Director (Macromedia)

Director is a powerful piece of authoring software which uses a time-based approach. An investigation showed that some very attractive, attention-grabbing multimedia had been created with this software, though its use seemed to be generally confined to the creation of linear programmes with somewhat limited interactivity. To some extent, this is natural for an authoring environment using a timeline rather than page-based paradigm.

The scripting language, Lingo, appeared to be quite powerful but was only poorly documented at the time of review. Also, only bitmapped graphics could be displayed; the vector artwork which proved so effective in CAL3P could not have been achieved with this package.

IconAuthor (AimTech)

This package was described as making professional authoring possible for non-programmers. This proved to be a reasonable claim; like Authorware, the graphical interface in which a presentation could be developed employed a network of nodes like a flow diagram, with each node having specific functions for the control of a multimedia presentation. It was possible to produce simple but attractive multimedia presentations such as might be required for public information kiosks; the scripting capabilities were largely inadequate for programming concepts other than simple quizzes or navigation through a presentation because nodes allowing more complex operations were not available.

'Open Database Connectivity' allowed IconAuthor to be linked with most common database packages. Features of this kind were becoming increasingly common, being found in ToolBook, Authorware and others. IconAuthor was unusual in that it was also available for Unix systems, a feature which might have been of interest since that platform was used by the CAD software in the DICTA project (see Chapter 4). Unfortunately, the Unix version did not have the full functionality of the Windows product, and for each medium, fewer formats were supported, so the software did not offer simple cross-platform compatibility.

Multimedia ToolBook (Asymetrix)

Like HyperCard, this authoring environment used a page-based concept. In fact, although the syntax of the two scripting languages was different, they offered virtually identical functions. Using ToolBook was an ideal way to build on the knowledge gained with the HyperCard prototype, making use of a wider range of media while also reaching the more common Windows compatible computers.

Linking ToolBook programmes to other applications via the Windows 'Object Linking and Embedding' protocol proved problematic, resulting in occasional, unpredictable crashes, but when that feature was disregarded few other problems were experienced. Within ToolBook, graphics were produced by drawing polygons, as in PowerPoint and some of the other basic slideshow programs – though bitmapped graphics in common formats could be displayed. It was possible to cut existing images from the DICTA PowerPoint slideshow and paste them straight into a sample ToolBook programme; in this way, an early version of the DIS software rapidly took shape, though more detailed diagrams were created later. Experimentation showed that the creation and modification of ToolBook's vector graphics could be scripted, so images could be modified in runtime, a key feature for CAL3P.

HiPWorks (Integrated Solutions Limited)

Investigation revealed that HiPWorks was essentially a program allowing easy construction of graphic user interfaces within an existing language which normally operated by command line interface. That language was Poplog, a combination of Pop-11, Prolog and Lisp. Unlike the scripting languages normally available within a piece of multimedia software, Poplog would allow object oriented programming or the construction of an expert system. Unfortunately, while Poplog was ideal for some tasks, the control of multimedia information was not one of them. Programming to handle user interaction and multimedia resources proved to be difficult. Some tasks which are simple in other packages were very difficult to achieve. This powerful scripting language was well suited to the storage and calculation which a highly interactive application such as CAL3P required, but HiPWorks was not satisfactory as a multimedia delivery system. In particular, it did not support hypertext, and could not display movies of any kind.

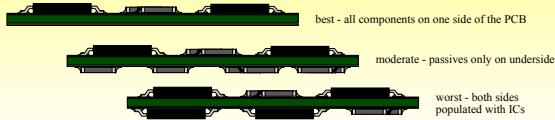
Appendix 3: Dicta Project Guidelines

Reproduced on the pages that follow are a number of slides from a PowerPoint slideshow which was originally created as a means of communicating the findings of the DICTA (Design for In-Circuit Testability Advisor) project.

As Chapter 4 describes, a set of principles had been identified which could reduce the incidence of manufacturing problems if PCB designers could be persuaded to heed them, and it was hoped that a multimedia approach would prove persuasive. The rules are described in detail in [Gallagher 1996].

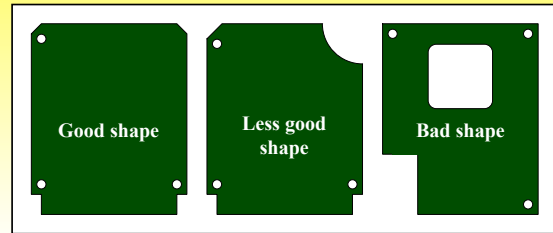
Component Mounting

- Preferably, all components will be on the top side of the circuit board
- Chip resistors and other small components can be placed on the secondary side of the board, but they must be located away from test pads
- Any flying wires should be located on the top side of the PCB, away from test probes



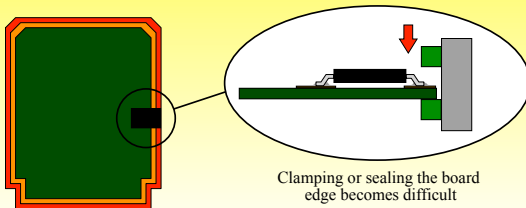
Board Shape

- To facilitate sealing for vacuum fixturing, boards should have straight sides and no internal cut-outs
- Any cut-outs must be surrounded by an area clear of all obstructions.



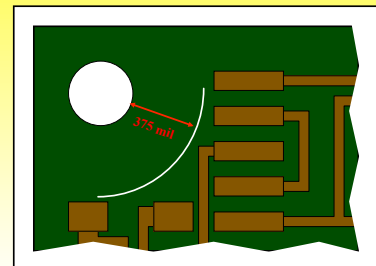
Edge Clearance

- An area around the board perimeter should be free of components, test locations and other features
- This should extend for at least 0.100, and preferably 0.200 inches from any edges or cut-outs



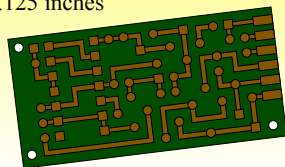
Tooling Hole Clearance

- The clear area around tooling holes should extend for 0.375 inches



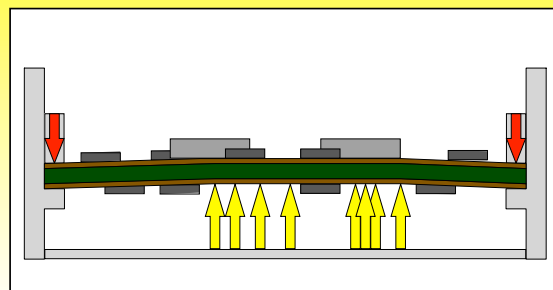
Locating Tooling Holes

- Tooling holes should be provided on at least two opposite corners
- If space allows, a tooling hole on a third corner will improve accuracy
- Holes should be offset so the board cannot be placed in a fixture the wrong way round
- All tooling holes must be unplated
- Minimum diameter is 0.125 inches



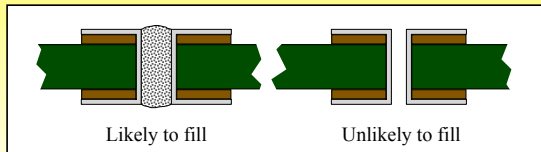
Board Thickness

- Thin board distorted during probing...



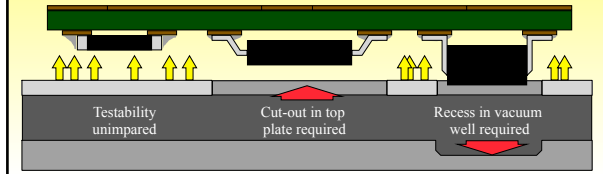
Via Holes

- To ensure a reliable vacuum seal between the assembled PCB and the test fixture, via holes should be of sufficient diameter (0.062 inches or more) so they will fill during flow soldering.



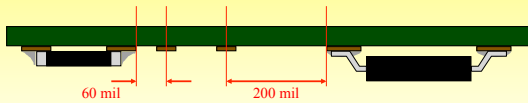
Component Height

- Components on the bottom side of the board must not obstruct the operation of the bed-of-nails fixture
- It is best if components have heights of less than 0.160 inches
- No components with heights greater than 0.360 inches should be mounted on the bottom of the board



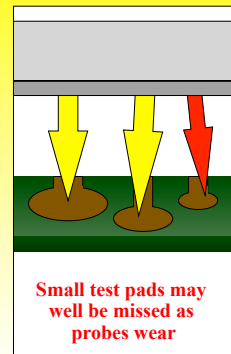
Component / Test Land Spacing

- The space between the centre of a test access point and the edges of any adjacent components must exceed 0.060 inches
- This distance must be increased to 0.200 inches where the height of adjacent components exceeds 0.160 inches



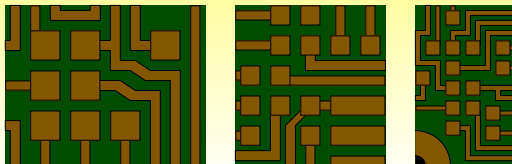
Test Land Size

- Test lands should be as large as possible to ensure reliable probing
- Recommended size is greater than 0.060 inches
- Minimum size is 0.035 inches for a bed-of-nails type fixture
- If the PCB requires two-sided probing, the minimum size for test lands on the top side is 0.040 because probing here is less accurate



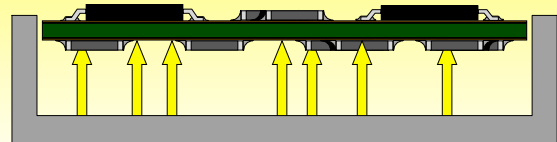
Land Spacing

- Test access points should be spaced with 0.100 inches between centres
- This can be reduced to 0.075 or even 0.050 where necessary
- The number of probes at smaller pitches should be kept to a minimum as these are less accurate



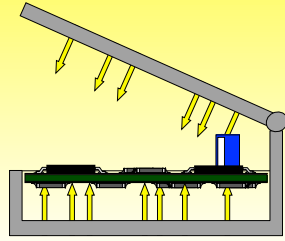
Board Probing

- If possible the board should be probed from one side only



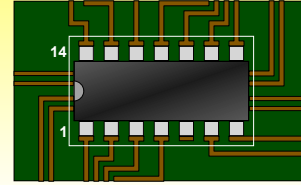
Board Probing

- If probing from both sides of the board is necessary it will be found that probing accuracy is reduced on the top side
- The location of tall components may also make testing difficult.



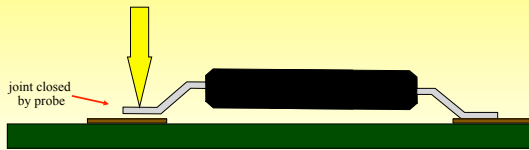
Component Orientation

- All packages of the same style should have the same orientation so that time will not be wasted trying to identify pins during manual testing
- Clear, permanent marking of pin numbering for each component is also necessary



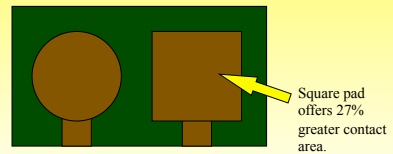
Probing Components

- Avoid probing the leads of the components
- Probing component leads can mask problems with bad solder joints
- Test pads should be included in the design instead



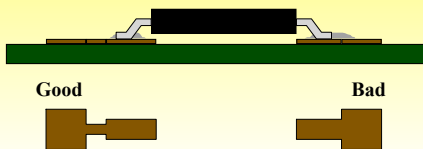
Test Land Geometry

- Square test pads offer a greater area to hit so the testing process is more reliable



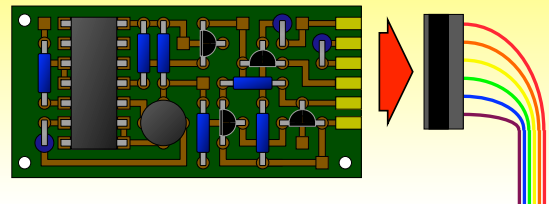
Test Land Geometry

- Solder on test pads may make probing difficult
- Mounting pads and test pads should be separated to prevent solder flowing onto the test pad during manufacture
- This also ensures the test probe will not strike a component skewed during placement or reflow



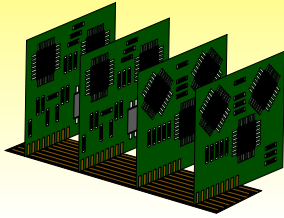
Edge Connectors

- In terms of testability, the best design for testing would achieve all the necessary contacts for testing at the edge connectors



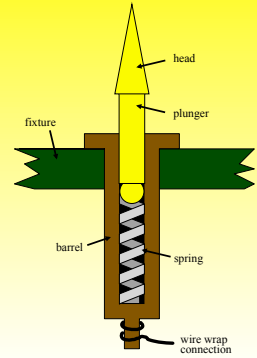
Modularity

- Ideally, complex products will be split into functional modules which can each be tested individually



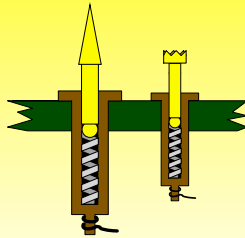
Test Probes

- Available in a variety of sizes and head configurations
- Larger probes have a longer reach but may not be practical in fine pitch applications



Test Probes

- A large probe should be used where possible for better electrical contact and improved fixture reliability.
- Smaller probes must be used where the test pads are closely packed or access is a problem.



Appendix 4: Student Exercises and Documentation

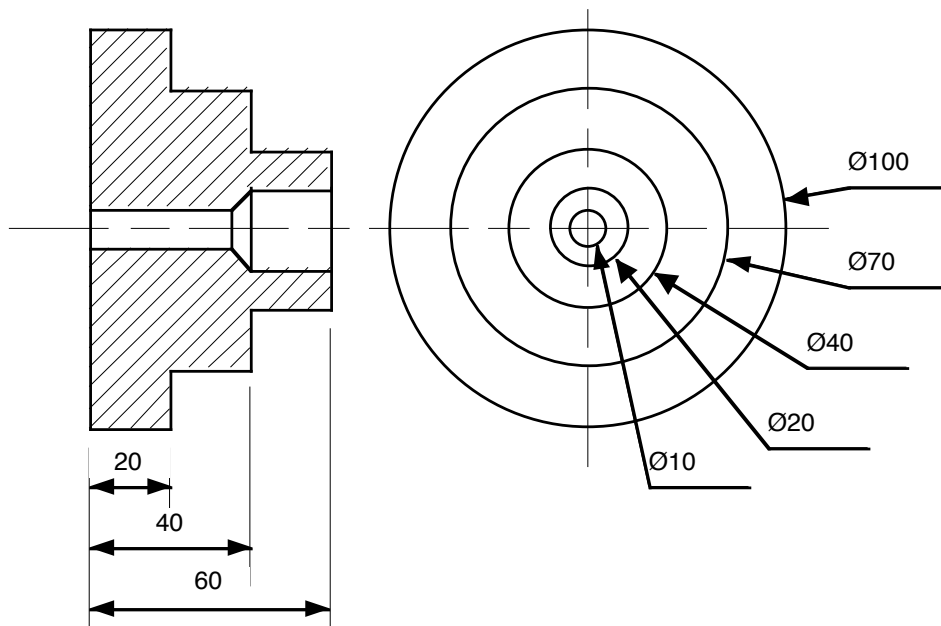
In each case, trials with the CAL3P software required students to work through a process planning exercise, which took the form of a part drawing. The exercises are reproduced here, together with the instruction sheet and evaluation form which students were also given.

The next page shows the process planning exercise which was employed with the HyperCard prototype. The exercise necessarily involves quite a simple workpiece because that version of the software did not support a full range of machining operations. The following page shows the evaluation form used in conjunction with the software at this stage. Results obtained were discussed in Section 5.3.2.

Once the Multimedia ToolBook version of CAL3P had been written, more complex exercises were possible, and a more detailed method of testing was necessary. The next page shows the process planning exercise used with the final form of CAL3P. In April 1997, the Windows version of CAL3P was operational but did not feature a built-in quiz; the written test which was employed at that stage is presented. Finally, there is the evaluation form which students used with the more modern versions of the software. Feedback received was discussed in Section 6.5.

Process Planning Exercise

A process plan is required to produce the pulley shown below on a CNC lathe. The material should be aluminium, which is available in 110mm bars. Use the interactive process planning tutorial to create a plan for the part. Your process plan should use a minimal number of setup operations, limit tool wear as far as possible and produce parts at a reasonable rate.



Process Planning Exercise – Your Comments

The interactive tutorial in process planning which you have used is still under development. Please help by commenting on your experience with the software.

Name: _____ (optional)

Consider the following statements, and circle one number where 1 = strongly disagree and 5 = strongly agree

- | | | | | | |
|---|---|---|---|---|---|
| 1. The built-in help function was useful. | 1 | 2 | 3 | 4 | 5 |
| 2. The exercise we were required to do was too easy. | 1 | 2 | 3 | 4 | 5 |
| 3. The way the program operated was clear. | 1 | 2 | 3 | 4 | 5 |
| 4. It was easy to understand how to define a process plan. | 1 | 2 | 3 | 4 | 5 |
| 5. The program adequately supports the theory we had learned. | 1 | 2 | 3 | 4 | 5 |

Do you feel that interactive tutorials of this nature are more enjoyable than conventional practical exercises?

(delete as appropriate) Yes / No

Do you feel that doing the interactive tutorial was more worthwhile than a conventional practical exercise?

(delete as appropriate) Yes / No

What were the best and worst things about the interactive tutorial?

What other features would you expect to see?

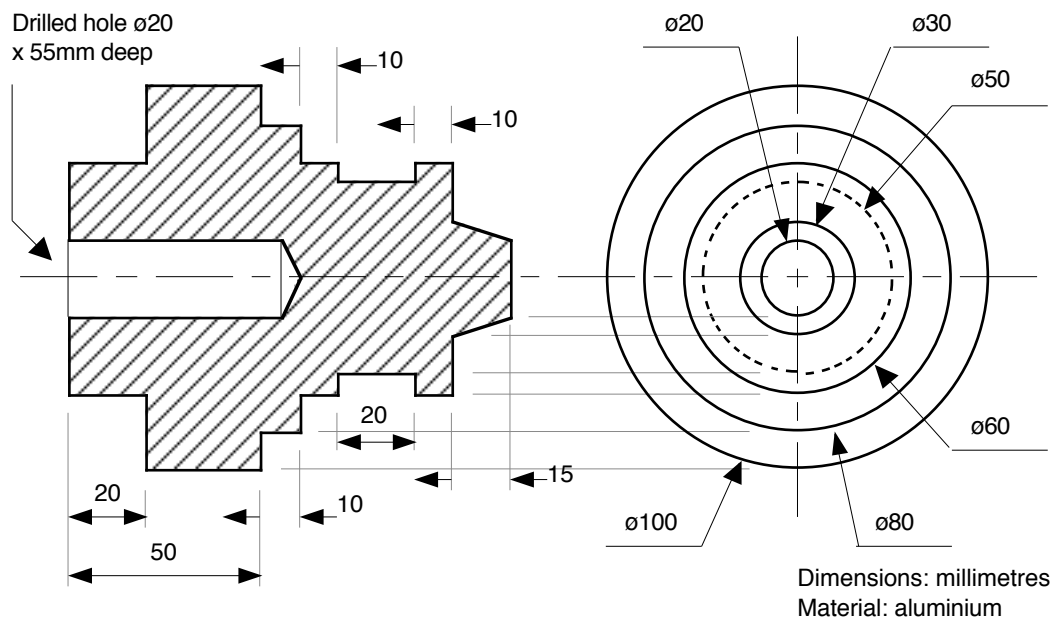
Any other comments:

Thank you for your assistance.

Process Planning Tutorial

Use the CAL3P software to write a process plan for production of the part shown below. It will be produced on a CNC lathe in occasional batches of 100. Aluminium suitable for the job is available in 105mm diameter bars which can be cut to any required length with a billet saw. A good process plan will use a minimal number of setup operations, limit tool wear as far as possible and produce parts of the right geometry at a reasonable rate.

When you have completed a process plan, test your work and make any changes necessary until you are satisfied. You should then do the short computer-based quiz which can be accessed from CAL3P's main menu.

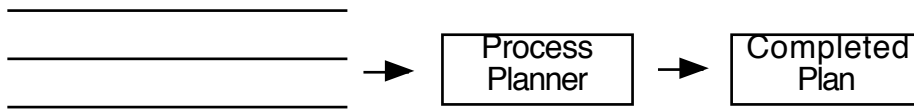


Process Planning Test

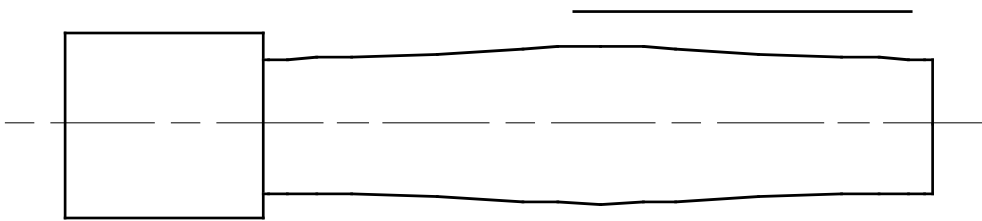
Name: _____
 Course: _____
 Date: _____

1. Identify the two other factors that the process planner must have in mind when writing a process plan.

eg: Part drawing

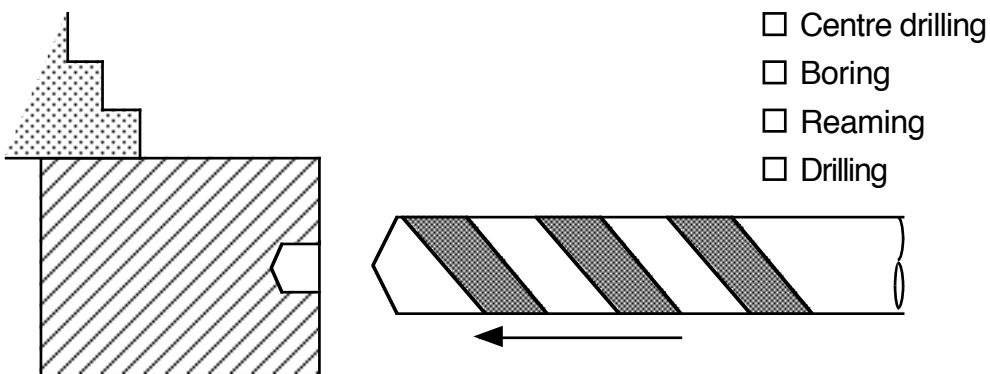


- 2a. Name the problem that has occurred in the manufacture of the turned part shown below (the 'bulge' is an undesired feature).



- 2b. If the problem with the turned part shown in question 2a happened when the workpiece was supported with a centre, what else might be done to solve the problem?.

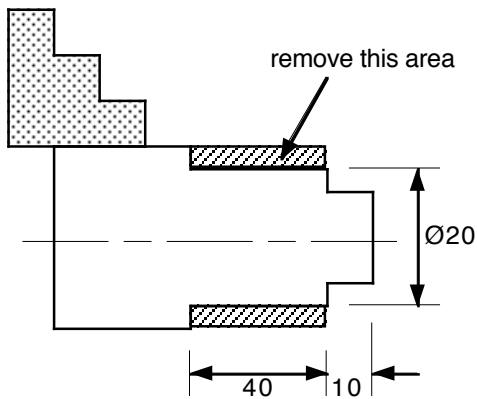
3. What name is given to the process shown in the diagram below? (Tick one answer)



4. What advantage is there to *turning between centres*, rather than simply gripping the workpiece in the chuck?

5. Write the process plan instruction which will remove the material shown in the diagram below:

2040

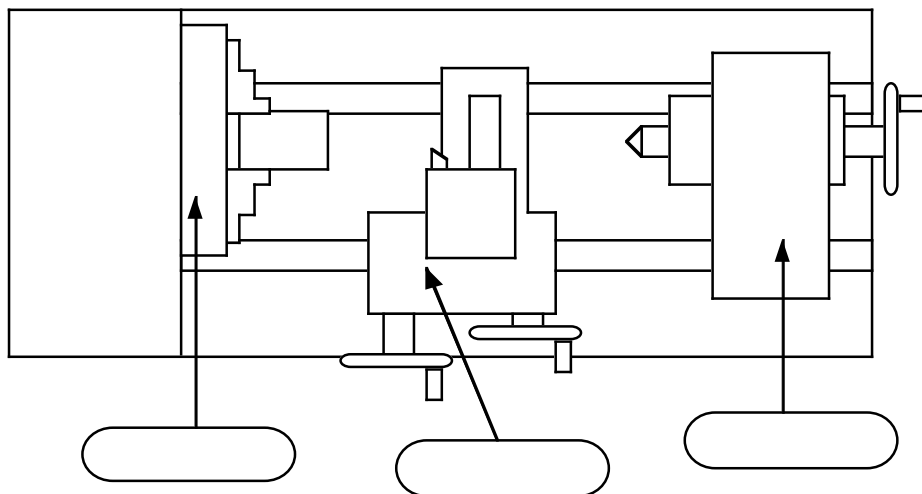


6. Most of the tools shown in CAL3P's photographs are tipped with removable tungsten carbide inserts. What do you think are the main reasons for the popularity of these?

7. Name some of the failure modes for a turning process if a very heavy cut is attempted:

eg: Workpiece

8. Add names to the three blank labels in this diagram of a typical centre lathe:



Your comments

Consider the following statements, and circle one number where 1 = strongly agree, and 5 = strongly disagree:

- | | (yes) | (no) |
|--|-----------|------|
| 1. The way to use the program was easy to understand. | 1 2 3 4 5 | |
| 2. CAL3P made writing a process plan easy. | 1 2 3 4 5 | |
| 3. The CAL3P exercise was more enjoyable than a normal tutorial. | 1 2 3 4 5 | |
| 4. It was easy to understand the syntax required for a process plan. | 1 2 3 4 5 | |
| 5. The program adequately supports the theory we had learned. | 1 2 3 4 5 | |
| 6. The photographs in CAL3P helped to understand turning principles. | 1 2 3 4 5 | |
| 7. The CAL3P exercise was more worthwhile than a normal lesson. | 1 2 3 4 5 | |
| 8. The diagrams in CAL3P helped to understand turning principles. | 1 2 3 4 5 | |
| 9. We were given plenty of time to use CAL3P properly. | 1 2 3 4 5 | |

What were the best and worst things about today's exercise?

What other features did you expect to see in CAL3P?

Any other comments?

Thank you for your assistance!

Appendix 5: A Model Answer to the Process Planning Exercise

Trials with CAL3P required students to complete a process planning exercise, reproduced in Appendix 4. Students' performance was measured, in terms of their marks in an on-line quiz, and also by the quality of their process plans. In order to judge their performance, it is necessary to understand what constitutes good practice; the plan presented in this appendix is one possible method for manufacturing the required part. A sample set of manufacturing instructions which would create the workpiece on page 247 is given below:

- 2000 Receive from stores, 100 off \varnothing 105mm x 160mm aluminium bars
- 2010 Grip in chuck, extension = 125
- 2020 Face off, depth = 2
- 2030 Straight turn, X = 0, L = 100, D = 100
- 2040 Straight turn, X = 0, L = 65, D = 80
- 2050 Straight turn, X = 0, L = 55, D = 60
- 2060 Grooving, X = 25, L = 20, D = 50
- 2070 Taper Turn, X = 0, L = 15, D1 = 20, D2 = 30
- 2080 Part off, X = 115
- 2090 Reverse and grip, extension = 50
- 2100 Straight turn, X = 0, L = 20, D = 60
- 2110 Centre drill, depth = 5
- 2120 Drill, diameter = 20, depth = 55
- 2130 Remove from chuck; end of process plan

When starting CAL3P, logging on as 'c3ptutor' puts a set of instructions like these into memory automatically, which might be useful for a quick demonstration, or for comparison with a student's own work. The remainder of this appendix describes the effect of each of the operations listed, identifying some of the alternative strategies which students used to achieve the manufacture of the part.

Firstly, it is necessary for the students to specify a suitably-sized bar. If the material is too small, it will be impossible to make the workpiece, while if they specify an oversized piece, time may be wasted removing large volumes of material. The sample process plan presented here is not ideal in this respect; some material is wasted, acting as a bar end which is gripped while the workpiece undergoes the initial processes, and later cut off. At this stage it is required that the bar must be gripped in the chuck in a safe

manner. The full length of the jaws should be used whenever possible, but there must be sufficient extension to allow access for subsequent machining operations. Figure 1 shows how CAL3P's display might appear at this point.

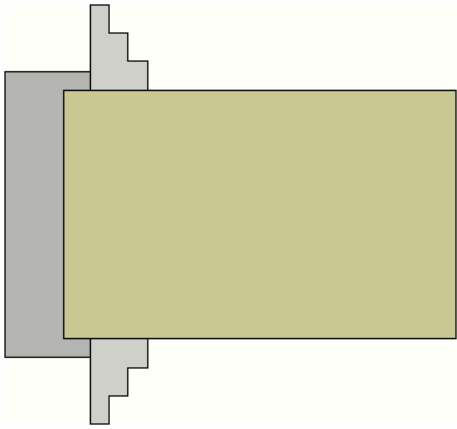


Figure 1: Clamping the bar material in the chuck

Next, it is desirable to carry out a facing operation, since the end of the bar may be rough sawn when supplied. A fine cut puts a good surface on the end of the bar, establishing a datum to work from (Figure 2).

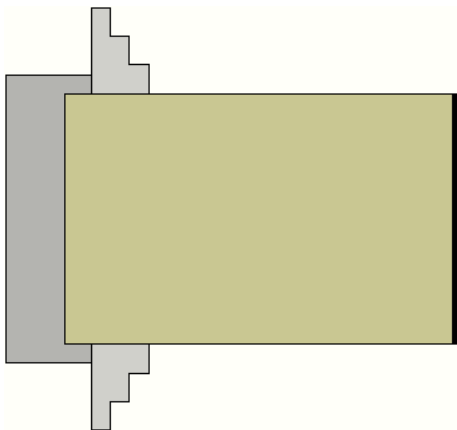


Figure 2: Facing off cut completed

Figures 3 to 5 show a sequence of straight turning operations which generate the 'stepped' features of the bar. Some students added a fourth straight turning operation to remove the bulk of the material in the vicinity of the tapered feature. This is a perfectly acceptable option.

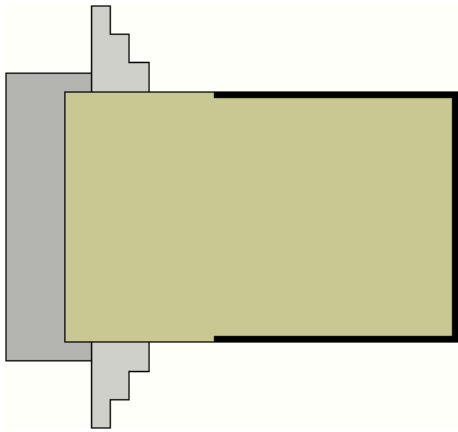


Figure 3: First straight turning operation

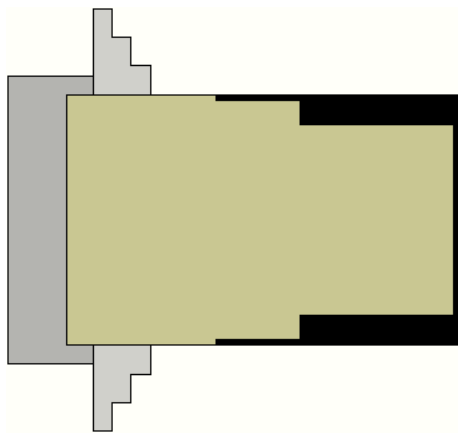


Figure 4: Second straight turning operation

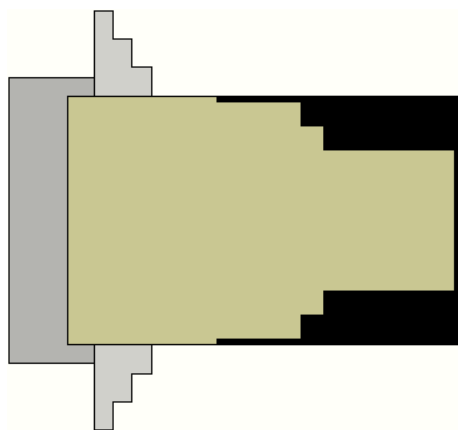


Figure 5: Third straight turning operation

Once the 60mm diameter has been established (by operation 2050 in the answer being discussed), it is possible to carry out a grooving operation to make the 50mm

diameter recess (Figure 6). At this stage, the tapered feature may also be added (Figure 7) – the sequence of these two operations is not critical.

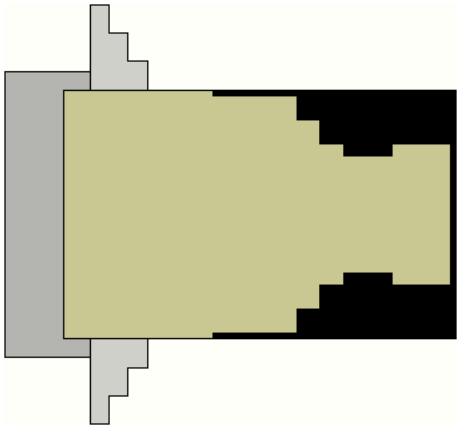


Figure 6: Grooving operation

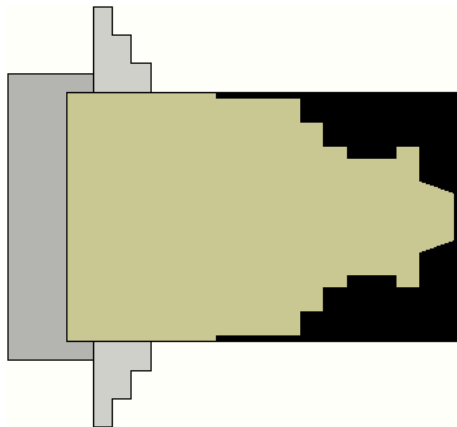


Figure 7: Taper turning operation carried out after grooving

Now all the operations which can reasonably be achieved from one end of the workpiece have been completed. The workpiece is now parted off from the rest of the bar (Figure 8). Some students may have specified a very long piece of bar material, allowing many parts to be brought to this stage.

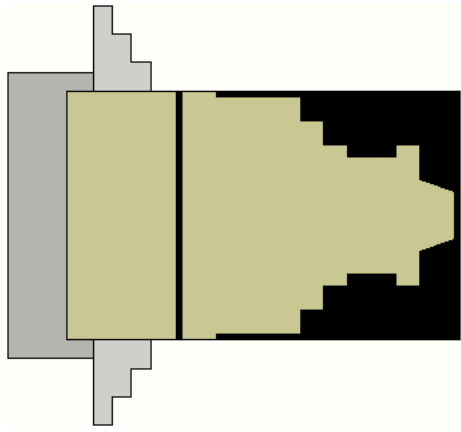


Figure 8: Parting off operation

Once the batch of 100 parts have all been brought to this stage, the final sequence of machining operations can begin. Each part is reinserted in the chuck, ideally clamped on one of the 'shoulder' features, to obtain a reasonable datum. Figure 9 shows how CAL3P's display would look at this point. Of course, any errors when supplying dimensions could have produce a workpiece with different geometry.

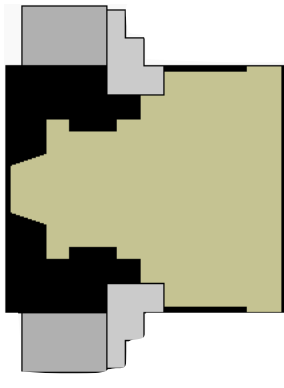


Figure 9: Workpiece is reversed and gripped again

The order of the remaining operations is not critical, except that centre drilling must precede drilling. The process plan lists the straight turning operation first (operation 2100); Figure 10 shows the workpiece after that stage.

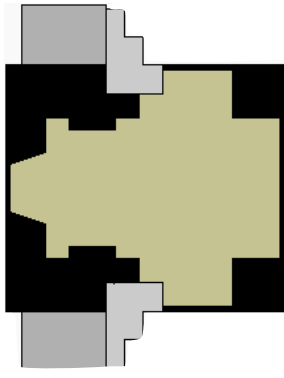


Figure 10: Straight turning operation

Figures 11 and 12 show how the screen display might look after centre drilling and drilling. These operations complete the manufacture of the workpiece, which can now be removed from the chuck, ending the process plan. Assuming a part of the right shape has been produced, this represents a satisfactory solution to the exercise.

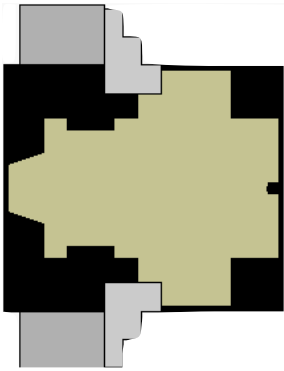


Figure 11: Centre drilling

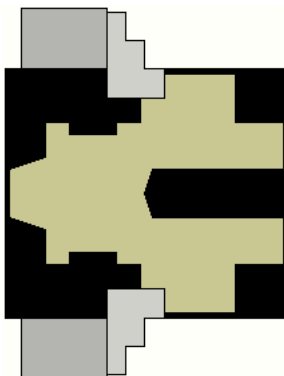


Figure 12: Drilling

Of course, a student may wish to carry out further experiments, either refining their existing process plan, or writing one for an entirely new part. Figure 13 shows some alternative parts, the manufacturing of which has been planned using CAL3P.

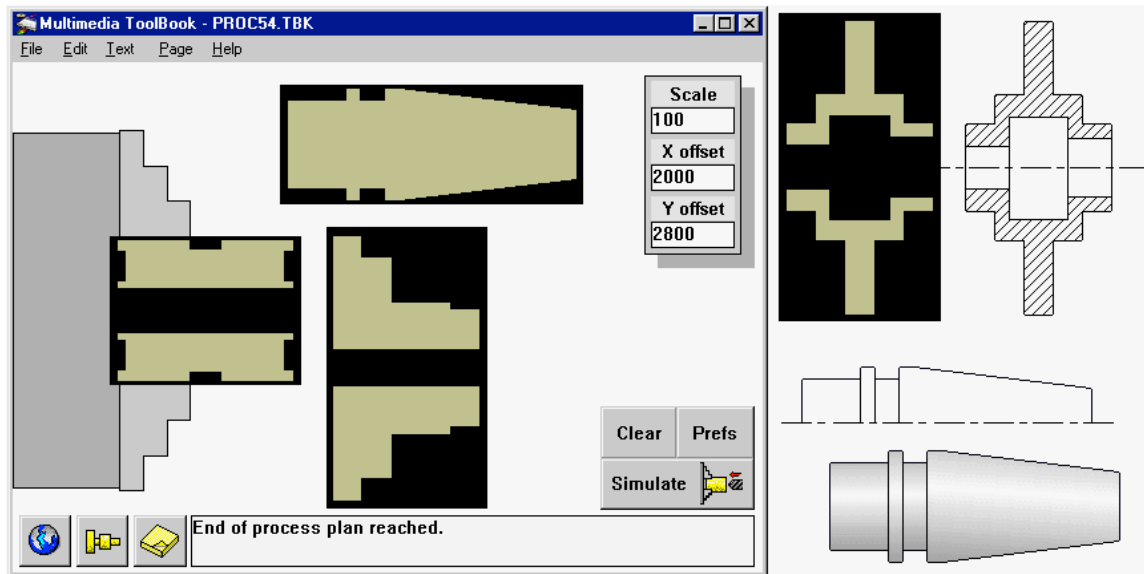


Figure 13: A selection of products 'machined' in CAL3P