

1996

Paradigms for the design of multimedia learning environments in engineering

Smith, Christopher Robert

<http://hdl.handle.net/10026.1/342>

<http://dx.doi.org/10.24382/4694>

University of Plymouth

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

**PARADIGMS FOR THE DESIGN OF MULTIMEDIA LEARNING
ENVIRONMENTS IN ENGINEERING**

by

CHRISTOPHER ROBERT SMITH

**A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of**

DOCTOR OF PHILOSOPHY

**School of Computing
Faculty of Technology**

September 1996

**PARADIGMS FOR THE DESIGN OF MULTIMEDIA LEARNING
ENVIRONMENTS IN ENGINEERING**

by

CHRISTOPHER ROBERT SMITH

**A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of**

DOCTOR OF PHILOSOPHY

**School of Computing
Faculty of Technology**

September 1996

Abstract

PARADIGMS FOR THE DESIGN OF MULTIMEDIA LEARNING ENVIRONMENTS IN ENGINEERING

CHRISTOPHER ROBERT SMITH

The starting point for this research was the belief that interactive multimedia learning environments represent a significant evolution in computer based learning and therefore their design requires a re-examination of the underlying principles of learning and knowledge representation.

Current multimedia learning environments (MLEs) can be seen as descendants of the earlier technologies of computer-aided learning (CAL), intelligent tutoring systems (ITS) and videodisc-based learning systems. As such they can benefit from much of the wisdom which emerged from those technologies. However, multimedia can be distinguished from earlier technologies by its much greater facility in bringing to the learner high levels of interaction with and control over still and moving image, animation, sound and graphics. Our intuition tells us that this facility has the potential to create learning environments which are not merely substitutes for "live" teaching, but which are capable of elucidating complex conceptual knowledge in ways which have not previously been possible. If the potential of interactive multimedia for learning is to be properly exploited then it needs to be better understood. MLEs should not just be regarded as a slicker version of CAL, ITS or videodisc but a new technology requiring a reinterpretation of the existing theories of learning and knowledge representation.

The work described in this thesis aims to contribute to a better understanding of the ways in which MLEs can aid learning. A knowledge engineering approach was taken to the design of a MLE for civil engineers. This involved analysing in detail the knowledge content of the learning domain in terms of different paradigms of human learning and knowledge representation. From this basis, a design strategy was developed which matched the nature of the domain knowledge to the most appropriate delivery techniques. The Cognitive Apprenticeship Model (CAM) was shown to be able to support the integration and presentation of the different categories of knowledge in a coherent instructional framework.

It is concluded that this approach is helpful in enabling designers of multimedia systems both to capture and to present a rich picture of the domain. The focus of the thesis is concentrated on the domain of Civil Engineering and the learning of concepts and design skills within that domain. However, much of it could be extended to other highly visual domains such as mechanical engineering. Many of the points can also be seen to be much more widely relevant to the design of any MLE.

Contents

	page
List of tables and illustrations.....	1
Acknowledgement.....	5
Author's declaration.....	6
1. Introduction, aims and overview.....	8
1.1. Aims.....	8
1.2. Overview.....	11
2. The subject matter of the MLE and its target learners.....	15
2.1. The subject matter.....	16
2.2. The target learners.....	20
3. Approaches to developing CBL: the need for a method.....	27
3.1. Styles of CBL.....	30
3.1.1. "Drill and practice".....	32
3.1.2. Tutorial.....	35
3.1.3. Intelligent tutoring systems.....	39
3.1.4. Hypertext/Hypermedia.....	48
3.1.5. Simulations.....	64
3.1.6. Microworlds.....	67
3.1.7. Comments on styles of CBL.....	70
3.2. Initial ideas for the development of the MLE.....	71
3.3. A brief review of some existing methods for CBL design.....	75
3.4. A method for the design of MLEs: the objectives of the research.....	81

4. Knowledge acquisition.....	83
4.1. Knowledge sources.....	83
4.2. The "Journeyman".....	86
4.2.1. Knowledge elicitation methods used.....	86
4.2.2. The example design problem.....	90
4.2.3. Description and characterisation of knowledge obtained.....	93
4.3. The "Master".....	101
4.3.1. Knowledge elicitation methods used.....	101
4.3.2. Description and characterisation of the knowledge obtained...	101
4.4. The "Industry" view.....	103
4.4.1. Knowledge elicitation methods used.....	103
4.4.2. Description and characterisation of the knowledge obtained...	103
4.5. Review of relevant models of the engineering design process.....	106
4.6. Methodological implications arising from knowledge acquisition.....	114
 5. Models of human knowledge: analysis of the information gained from knowledge acquisition.....	117
5.1. Symbolic or rule-based.....	118
5.2. Experiential or pattern-matching.....	122
5.3. Metacognitive or task-model-based.....	129
5.4. Summary.....	130
5.4.1. Knowledge representation paradigms.....	130
5.4.2. Implied most suitable presentation styles.....	133
 6. Pedagogic models: the requirement for a pedagogic model.....	136
6.1. Models based on Instructional-Design theory.....	136
6.2. Models developed from Cognitive Psychology and Constructivism....	138

7. The instructional basis of the MLE.....	146
7.1. An overview of the Cognitive Apprenticeship Model (CAM).....	146
7.2. How the identified knowledge representation paradigms and their associated knowledge types map onto the categories of knowledge recognised by the CAM.....	159
8. Design and implementation of the MLE.....	165
8.1. Illustrated schematic overview of the MLE.....	166
8.2. Underlying design of the MLE: mechanisms needed to realise the major elements of the CAM.....	171
8.3. How the human tutor may use the MLE: inherent limitations of the MLE.....	175
8.4. Authoring systems used and technical problems encountered.....	179
8.5. Formative Evaluation.....	183
9. Conclusions.....	187
9.1. Overview.....	187
9.2. Methodological perspective.....	188
9.3. The Knowledge Engineering approach.....	191
9.4. The Cognitive Apprenticeship Model	193
9.5. Possible directions for improving the design and implementation of the MLE using newly available technology.....	195
9.6. Critique, reflection and the direction of future work.....	198
Abbreviations used.....	201
Appendix I: Solution to the elicited example design problem.....	202
References.....	219

List of tables and illustrations

	page
Figure 1. An example of a Dam, Spillway and Energy Dissipator (Shiroro Hydroelectric Project, Nigeria, Water Power & Dam Construction, November 1986, page 14).....	16
Figure 2. Erosion at the toe of a spillway (Elevatorski 1959, page 207).....	17
Figure 3. Diagram of a United States Bureau of Reclamation Type I Hydraulic Jump Stilling Basin, adapted from a diagram of a Type IV Basin (US Bureau of Reclamation, 1987, page 390).....	18
Figure 4. Video still of the side view of the formation of an hydraulic jump (video filmed in a laboratory by the author).....	18
Figure 5. Diagram of a Submerged Slotted Bucket Dissipator (adapted from a diagram in: US Bureau of Reclamation, 1987, page 398).19	19
Figure 6. Diagram illustrating the energy dissipating action of a Submerged Slotted Bucket (US Bureau of Reclamation, 1987, page 398).....	19
Figure 7. Organisation of physical structures.....	94
Figure 8. Organisation of physical structures including attributes.....	96
Figure 9. Example equation: Incoming velocity at hydraulic jump stilling basin floor level.....	97

Figure 10. Example environmental data: General storm dimensionless S-graph data, Rocky Mountains, describing the actual flows following rainfall.....	98
Figure 11. Simplified pseudo code example of a procedural confirmation... ..	99
Figure 12. Simplified pseudo code example of a problem control strategy... ..	100
Figure 13. Mapping between the social model and the technical model via the design schema. (Externally imposed requirements and constraints are mapped onto the technical model of the problem by the design schema which in turn controls the search for technical solution concepts), (Ball, 1990).....	110
Figure 14. "Stills" of a hydraulic jump taken from a video filmed in a laboratory by the author.....	125
Figure 15. Explicit Metacognitive knowledge: a PERT network.....	126
Figure 16. "Stills" taken from an animated simulation of the Hydraulic Jump created by the author.....	134
Figure 17. How the knowledge representation paradigms of the domain of dam spillway design map onto the types of knowledge recognised by the CAM.....	163
Figure 18. MLE Architecture.....	166
Figure 19. Example screens from the "Coach".....	167

Figure 20. Example screens from a “Book”.....	169
Figure 21. Example screen from the “Library”.....	170
Figure 22. A generic design problem decomposed into a number of tasks...	171
Figure 23. Design problem nodes with example difficulty ratings and sequence rankings.....	172
Figure 24. Decomposition of the tasks required to produce a preliminary design for a hydraulic jump stilling basin.....	174
Figure 25. Example design problem controlling dimensions and tailwater conditions (based on Figure 9-51, page 402, US Bureau of Reclamation (1987)).....	203
Figure 26. Variables relating to the equation to determine stilling basin depths.....	205
Figure 27. Nomograph used to determine stilling basin depths (based on Figure 9-43, page 396, US Bureau of Reclamation (1987)).....	206
Figure 28. Variables used to help select the required type of stilling basin...	208
Figure 29. Dimensions for a type III hydraulic jump stilling basin (taken from Figure 9-41, page 393, US Bureau of Reclamation (1987)).	210
Figure 30. Nomograph used to fix the length of a US Bureau of Reclamation type III stilling basin (based on Figure 9-41, page 393, US Bureau of Reclamation (1987)).....	211

Figure 31. Submerged buckets and their hydraulic action (taken from Figures 9-45 and 9-46, pages 398 and 399, US Bureau of Reclamation (1987))..... 212

Figure 32. Variables used in the design of a submerged bucket dissipator (taken from Figure 9-50, page 402, US Bureau of Reclamation (1987))..... 213

Figure 33. Nomograph for the design of a slotted submerged bucket dissipator (taken from Figure 9-49, page 401, US Bureau of reclamation (1987))..... 216

Figure 34. Dimensions for a submerged slotted bucket dissipator (taken from Figure 9-45, page 398, US Bureau of Reclamation (1987)). 218

Acknowledgement

I would like to thank Dr Peter Jagodzinski (Reader in Human Computer Interaction at the University of Plymouth), my Director of Studies, for always readily giving his support, enthusiasm and invaluable guidance.

I would also like to thank, for their vital specialist assistance,

my secondary supervisors:

Dr Ian Parmee

Senior Research Fellow and Manager of the Engineering Design Centre at the University of Plymouth

Professor Geoff Bullock

Tarmac Professor and Head of Dept. of Civil & Structural Engineering at the University of Plymouth

Mike Phillips

Senior Lecturer and Course co-ordinator for the MediaLab Arts Degree at the University of Plymouth

my external contacts:

Mike Stancombe

Training Manager of the ICE

Paul Hayes

WS Atkins (Plymouth)

This research has been funded by an EPSRC Research Studentship.

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award.

This study was financed with the aid of a studentship from the Engineering and Physical Sciences Research Council.

Relevant scientific seminars and conferences were regularly attended at which work was often presented; external institutions were visited for consultation purposes, and several papers prepared for publication.

Publications:

"The Implementation of a Multimedia Learning Environment for Graduate Civil Engineers" co-written with Peter Jagodzinski. Published in the Association for Learning Technology Journal, Volume 3, Number 1, 1995 (Selected Papers from the Association for Learning Technology Journal Conference 94).

Paper in preparation:

"The Use of Multiple Paradigms for the Representation of Knowledge in a Multimedia Learning Environment for Engineering Design" co-authored with A.P. Jagodzinski and I.P. Parmee, to be resubmitted to the International Journal of Human-Computer Studies.

Presentations and Conferences Attended:

European Society for Engineering Education Conference 92, Nottingham, 26-28th March 1992. Presented a poster outlining my area of research.

European Society for Engineering Education Conference 94, Vassa, 8-11th June 1994. Had a paper: *"A Knowledge Engineering Approach to the Design of a Multimedia Learning Environment"* co-written with A.P. Jagodzinski and I.P. Parmee, presented by A.P. Jagodzinski.

Association for Learning Technology Conference 94, Hull, 19-21st Sept 1994. Presented a paper: *"The Implementation of a Multimedia Learning Environment*

for Graduate Civil Engineers" co-written with A.P. Jagodzinski.

Association for Educational & Training Technology International Conference 95, Plymouth, 10-12th April 1995. Research formed part of a workshop: "*Models of Knowledge, Learning and Representation for Multimedia Learning Environments.*"

Presented yearly at Departmental research seminars.

Courses Attended:

Attended lectures given by Peter Jagodzinski on a course entitled "*Multimedia Teaching and Learning*" taken by year 2 undergraduates on the MediaLab Arts degree.

External Contacts:

Mr Mike Stancombe, Training Manager of the Institution of Civil Engineers (I.C.E.).

Mr Paul Hayes of WS Atkins, Plymouth

Signed C.R. Smith
Date 8th Nov 97

1. Introduction, aims and overview

The functions of this chapter are:

- (a) *to describe a need for the design of multimedia learning environments to be based on new and reinterpreted principles rather than existing CBL approaches*
- (b) *to declare the specific aims of the research*
- (c) *to provide an overview of the main phases of the research programme*

1.1. Aims

The research undertaken aims to contribute to a better understanding of the new ways in which computer based multimedia learning environments (MLEs) can aid learning, and to provide a design method which reflects that improved understanding.

The research was motivated by a belief that recent developments in the technology of interactive multimedia has the potential to revolutionise computer based learning. This technology offers much greater facility in bringing to the learner high levels of interaction with and control over still and moving image, animation, sound and graphics. Our intuition tells us that this facility has the potential to create learning environments which are capable of elucidating complex conceptual knowledge in ways which have not previously been possible.

This thesis examines the underpinning theory, knowledge acquisition, analysis, design and partial implementation of a MLE for civil engineers who are undergoing training in the design of hydraulic structures. However, the findings of

the research are intended to be generic to the field of learning through multimedia.

The term "Multimedia Learning Environment" as used in this work means an interactive, computer-based system in which the learner has access to a wide variety of media about a particular topic, structured to provide a learning experience with identified objectives, including practical exercises and assessment. Typically such an environment would support up to two hours of student effort on a topic, not counting repetitions. The MLE would normally form one or more small parts of a larger programme of learning, for example a one semester module in a degree course, and be combined with other forms of teaching such as, live tutorials, lectures, practicals, site visits, project work, distance learning material and so on. The value of the MLE would be to enable self-paced, student-centred learning to take place in support of other learning methods.

Underlying the approach taken to the development of the MLE was the proposition that conventional approaches to analysing learning needs and designing computer based learning systems arise from older, relatively limited technologies. As such they are both incapable of capturing the richness of knowledge present in complex learning domains such as civil engineering and unable to take full advantage of the potential of the newer technology of computer based interactive multimedia.

Furthermore, adherence to conventional approaches can restrict the range of learning theory considered and the corresponding learning needs analysis and design methods that are used. The author proposes that in order to capture the richness of knowledge present in a complex learning domain, such as civil engineering, to the greater extent that is made possible by recent developments in computer based interactive multimedia, a design method should be used that

is not prematurely constrained by an early choice of style of delivery or teaching model. Such a method would require:

- analysis of target learners in terms of the learning goals, environment and characteristics
- knowledge elicitation in the problem domain from the relevant range of experts
- detailed analysis of the knowledge elements in order to identify the mechanisms of human learning which suit them best
- identification of the multimedia techniques which most readily facilitate the types of learning required
- selection of a coherent instructional model which supports the integration of the different techniques within a holistic pedagogic strategy
- development of a software architecture to enable the design to be realised
- design of a suitable and effective user interface

The scope of the work is confined to these aspects of the design of MLEs and does not extend to production or evaluation issues.

The research was concerned with developing and evaluating this approach. Its value would be judged in terms of the contribution made by the design approach towards enabling designers to improve the richness and effectiveness of the learning experience.

1.2. Overview

The research, following the steps outlined above, involved an initial process of learning needs analysis based on techniques from the field of knowledge engineering. Information was elicited from different categories of expert to acquire a wide range of problem domain knowledge. The acquired domain knowledge was then analysed in detail and characterised by being mapped onto a number of different paradigms of human knowledge representation. Each of these paradigms in turn implied their own styles of teaching and knowledge presentation.

After learning needs analysis an instructional model was selected that was capable of conveying, within a structured framework, all the acquired types of knowledge according to the most appropriate representational paradigms. The instructional model was adaptable enough to be implemented on a computer and comprehensive enough to cover the domain. This enabled the design and implementation of the MLE to be explicitly based on the model.

Finally, key aspects of the design were implemented and evaluated.

A number of benefits were gained from using this approach to design. First, it led to the recognition that in this domain and probably others, different categories of expert exist, each with their own perspective and priorities. The inclusion of these multiple views can enrich learning in important ways. Secondly, analysis of the domain knowledge in terms of its human learning mechanisms and underlying representational paradigms brought to light qualities and categories of knowledge that were not evident at face value. This greater awareness of the knowledge has enabled the designers to capture and express aspects of the domain which would otherwise probably have been missed. Thirdly, once different categories of domain knowledge were understood in terms of human

learning it became possible to see how interactive multimedia could be used to convey knowledge in forms most sympathetic to the learner. Fourthly, these steps provided a basis for selecting an instructional model for the architecture of the MLE which was capable of integrating and delivering the different types of knowledge in their most appropriate representation styles.

The drawbacks of the approach arose from the additional work required in the early stages of the design process, including the need for the designer to become familiar with substantial bodies of theoretical knowledge. However, as in most fields, greater effort in the early stages of design can be seen to yield significant improvements, both to the clarity of requirements for production and to the quality of the finished product.

In more detail the full sequence of the work done for the research, as explicitly set out in the chapters of this thesis, involved:

Chapter 2. Identifying a domain and target learners for the MLE which would act as a vehicle for the research.

Chapter 3. Reviewing examples of different styles of educational software, the development methods used to produce them and the learning theories they are based on. Stating the objectives of the research.

Chapter 4. Eliciting domain knowledge from different categories of domain expert. Briefly examining models of the engineering design process to gain further insight into the structure of the domain material to be taught by the MLE.

- Chapter 5. Analysing the acquired knowledge: characterising the knowledge by interpreting it in terms of a number of different human learning mechanisms and knowledge representation paradigms.
- Chapter 6. Reviewing pedagogic models to identify a pedagogic framework and instructional model capable of integrating all the categories of acquired knowledge and presenting them in their most natural style.
- Chapter 7. Selecting the Cognitive Apprenticeship Model (CAM). Defining the way in which the domain knowledge would be supported within the CAM by the MLE.
- Chapter 8. Developing a unifying architecture and detailed techniques with which to implement the MLE. Designing the presentation layer which represents the MLE to the user and implementing and evaluating with users key components of the MLE.
- Chapter 9. Analysing the benefits and costs of the design method and identifying the elements which make a useful contribution to the methodology of MLE design.

At various stages during the research, new areas of theory and practice emerged which had to be investigated in order to determine in which direction to proceed. The literature reviews generated from these investigations are not collected together at the beginning of the thesis. Instead, they appear throughout the work at the points where they were undertaken and where they make the most sense in terms of what follows. For instance, a review of possible instructional models on which to base the MLE could only take place (and would only be properly appreciated when being retold) after the learner and learning domain had been

identified and all the relevant domain knowledge had been acquired and analysed in detail.

2. The subject matter of the MLE and its target learners

This chapter describes the particular learning domain and set of target learners which are used as a vehicle to pursue the aims of the research. The domain was chosen for the visual richness of its theoretical and practical content. The target learners are in transition from theoretical, undergraduate learning to professional practice.

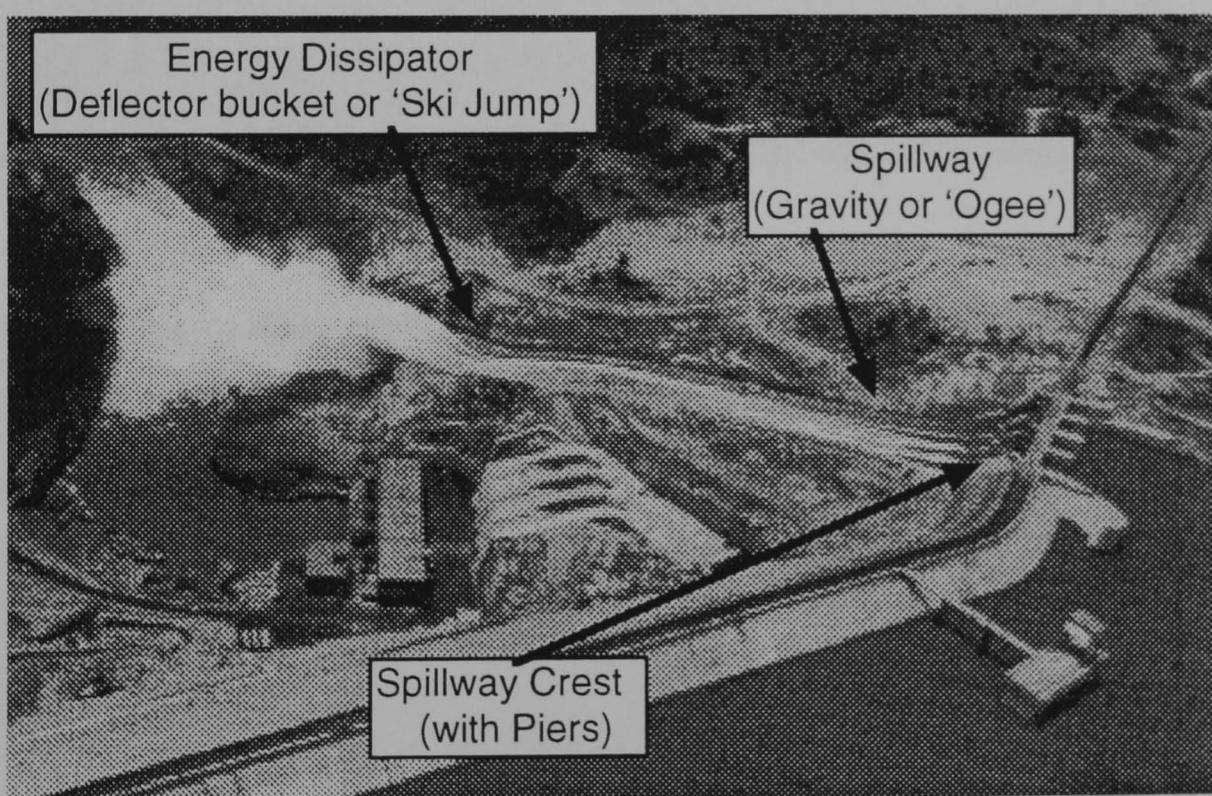
Both domain and learners should be regarded as exemplifying wider, generic issues within the design of systems to support learning.

This chapter first provides a short overview of the subject matter of the MLE. Its purpose is to set the context of the taught material and to introduce the terms and concepts which are covered in detail in later chapters. Secondly, it provides profiles of the target learners in terms of their prior knowledge and learning goals. These details are necessarily highly specific to one domain. However, it should be remembered that they are only intended to illustrate and exemplify generic principles and approaches.

2.1. The subject matter

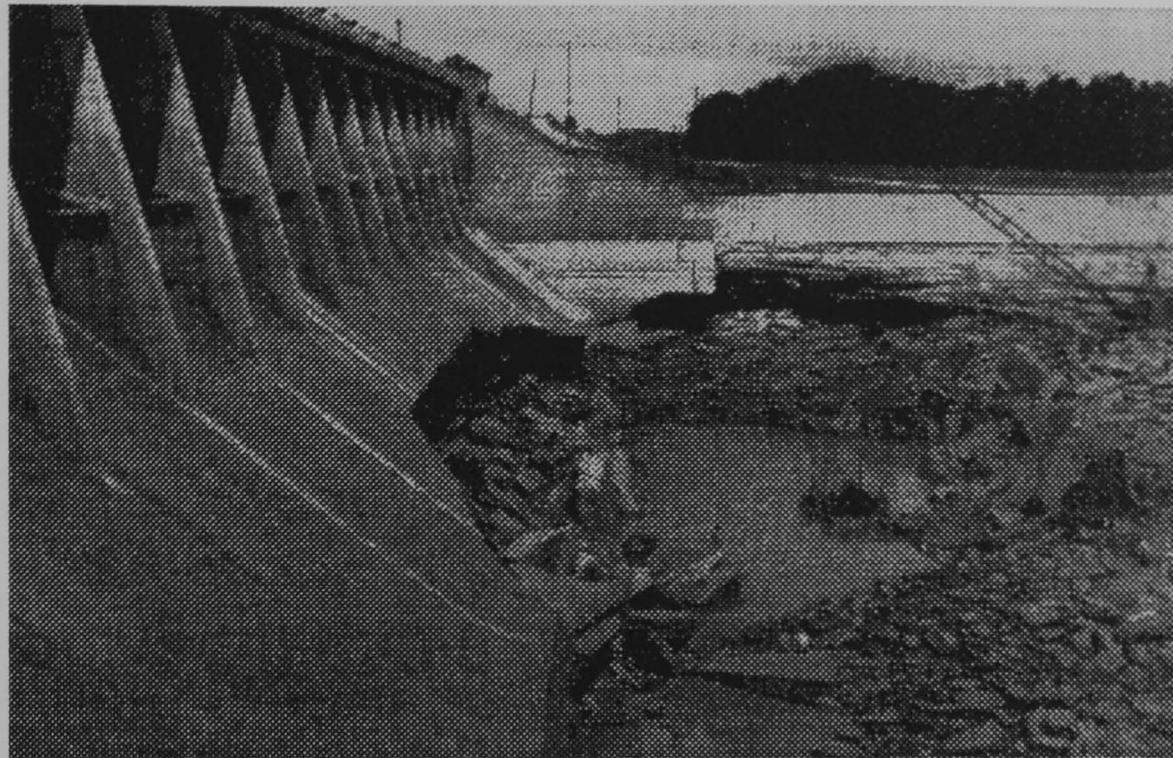
The subject matter used as a vehicle for the research concerned the problem of selecting and dimensioning energy dissipating structures for dam spillways. An example of an energy dissipating structure is shown, in operation, in Figure 1.

Figure 1. An example of a Dam, Spillway and Energy Dissipator (Shiroro Hydroelectric Project, Nigeria, Water Power & Dam Construction, November 1986, page 14).



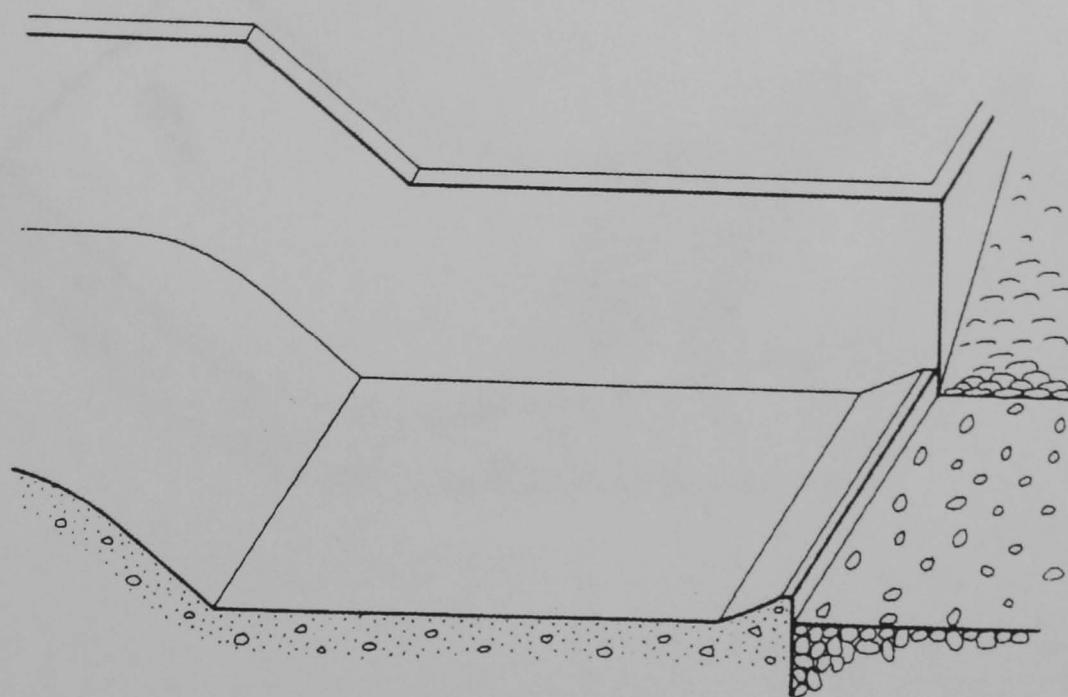
Spillway energy dissipators or terminal structures are used to return the shallow, fast moving water flowing down a spillway to the depth and velocity appropriate to the river below the dam. Without such a structure severe erosion at the toe of the dam could occur. An example of erosion occurring at the base of a dam is illustrated in Figure 2.

Figure 2. Erosion at the toe of a spillway (Elevatorski 1959, page 207).



In particular, the subject matter used concentrated on the theory behind and the design of hydraulic jump stilling basins (see Figures 3 and 4) and submerged bucket dissipators (see Figures 5 and 6).

Figure 3. Diagram of a United States Bureau of Reclamation Type I Hydraulic Jump Stilling Basin, adapted from a diagram of a Type IV Basin (US Bureau of Reclamation, 1987, page 390).



The basin is used to safely form an hydraulic jump (see Figure 4).

Figure 4. Video still of the side view of the formation of an hydraulic jump (video filmed in a laboratory by the author).



An hydraulic jump provides an effective means to dissipate the energy of the water flowing down a spillway and return the flow to the velocity and depth suited to the river below the spillway.

Figure 5. Diagram of a Submerged Slotted Bucket Dissipator (adapted from a diagram in: US Bureau of Reclamation, 1987, page 398).

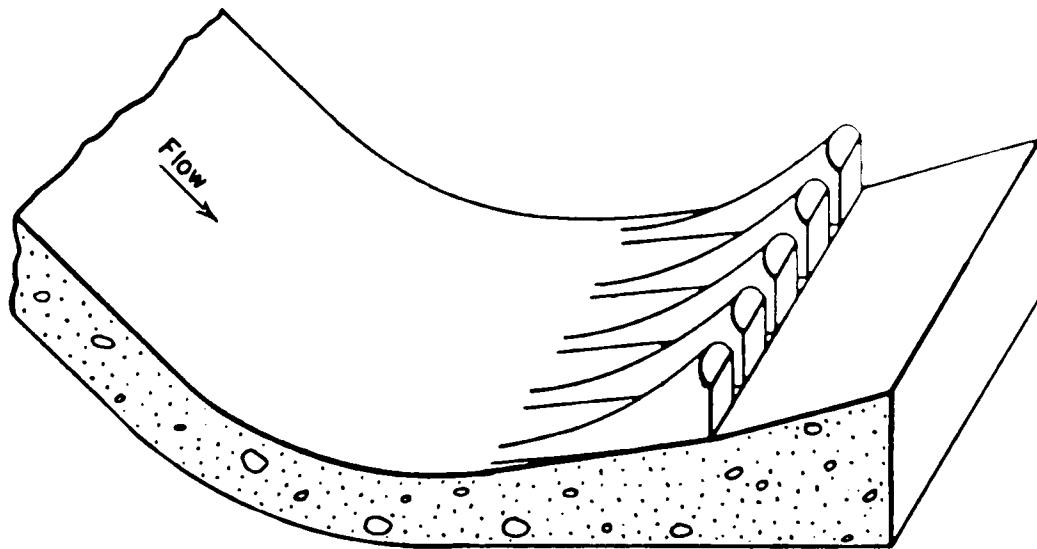
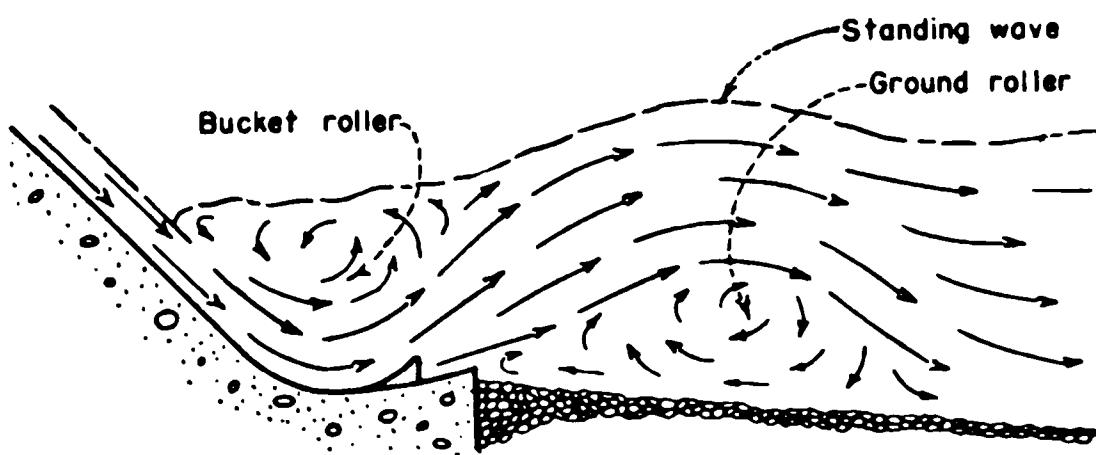


Figure 6. Diagram illustrating the energy dissipating action of a Submerged Slotted Bucket (US Bureau of Reclamation, 1987, page 398).



Submerged Bucket dissipators are typically used when the depth of water at the base of the spillway is too great for the formation of an hydraulic jump.

2.1. The target learners

(a) trainee graduates

The MLE is aimed at supplementing and supporting the inservice training that is given, under the supervision of a Training Manager (termed Supervising Civil Engineer, see below), to trainee graduates who are working within a civil engineering practice. A graduate has to undertake several years of professional training in order to achieve chartered membership of the Institution of Civil Engineers (I.C.E.).

The process and requirements necessary to become a chartered member of the I.C.E. are detailed in the I.C.E. document: "I.C.E. 101 (Winter 1992), Routes to Membership: Chartered Engineer". The document provides a general overview of the usual working and training context of a graduate civil engineer.

The document states that, amongst other factors, to become a chartered Civil Engineer a graduate with an accredited degree (such as a BEng (Hons) in Civil Engineering from the University of Plymouth, discussed in section (b)) senior graduates must:

- (i) have achieved the Institution's Core Objectives;
- (ii) have undertaken a programme of continuing education;
- (iii) have had responsible practical experience in one of the many branches of Civil Engineering.

The Institution strongly recommends that candidates should achieve their Core Objectives by entering a formal Training Agreement whereby their training is structured to meet Core and Specific Objectives, and is carried out under the guidance of a Supervising Civil Engineer (SCE) through an employer on the

Index maintained by the Institution. The Specific Objectives are drawn up by SCEs on behalf of the trainees' employing organisations and are intended to cover the requirements and practices of the organisations and the job-related needs of the trainees.

Candidates undergoing this Approved Training may expect to take between two and four years to meet the Core and Specific Objectives. Completion is verified by means of the Training Review.

A schedule of the objectives is formed by combining the Core and Specific Objectives and is intended to provide the basis for ensuring that trainees achieve the level of competence required by the Institution before applying for a Training Review. The schedule has three components:

- (i) professional and general;
- (ii) the engineering solution process; and
- (iii) the implementation process

The engineering solution and implementation process components of the schedule are intended to develop the trainees' ability to apply their academic knowledge and subsequent training to the solution of practical civil engineering problems of some complexity.

The candidate may submit themselves to the Chartered Professional Review (CPR) provided they have:

- (a) passed the Training Review;
- (b) gained practical experience in a position of responsibility; and
- (c) undertaken at least 30 days' continuing education

The CPR takes the form of a rigorous review of their career to date to verify that their education, training and experience will enable them to discharge in full the responsibilities of a Chartered Engineer. On successful completion of the CPR candidates are admitted to Corporate Membership (MICE) of the Institution and gain the Chartered Civil Engineer (CEng) status which this confers.

The review procedures are designed to enable candidates to demonstrate that during their employment they have:

- (i) developed and proved their technical and professional competence including the exercising of independent judgement requiring both responsible experience and the application of engineering principles; and
- (ii) acquired an understanding of financial, commercial, statutory, safety and environmental considerations.

The 30 days of continuing education may include courses, technical conferences, seminars, symposia, private studies of a structured nature, organised site visits and meetings of professional bodies.

The evidence used to reach a decision about whether or not candidates have reached the required standard comes from a number of sources including a submitted Training Record for candidates who have completed any period of Approved Training and a submitted record of Continuing Professional Development which details all the candidates' continuing education activities as certified by the SCE.

The Training Record must be kept by all trainees under agreement and shows:

- (i) the approved Training Scheme with achievements certified by the SCE (who will set tasks, monitor progress and direct further work);

- (ii) the trainees' quarterly reports on the practical experience that has been gained; and
- (iii) record of involvement in Institution affairs.

The above outline presents a picture of the working and training development of a graduate civil engineer as being a well planned and structured process that is monitored by a SCE and verified by a Professional review. Typically, a graduate trainee is attached to design project teams and is required to undertake specific small design sub-tasks, such as the design of a spillway terminal structure. They would be expected to present, explain and justify their designs to the project leader and other members of the team. The graduate's work and training exists within a professional expert culture within the employing company where they can receive expert assistance and guidance from their SCE. Also, as graduates work on design problems they can access information from the company's stock of material on previous projects such as working drawings, specifications and designs.

In general, the proposed MLE was an attempt to supplement and assist a graduate's training by providing tailored access to interactive multimedia information (some of which might typically be provided during their 30 days of continuing education), for example, design problems, coaching material, animations, videos, assessment and explanations. Some of the expert assistance, available within a company, was incorporated to help in several key aspects of the solution of a design problem. The MLE was designed so that, to some extent, a trainee's progress while using the MLE could be monitored and controlled by their SCE.

Details of some of the specific skills and knowledge that were thought to be needed by a new Civil Engineering graduate are examined in 'Chapter 4. Knowledge acquisition'. Chapter four covers the elicitation, from 3 categories of

expert, of an essential, representative set of skills and knowledge which it was felt the MLE could help to convey.

The training offered by the MLE can be seen to relate to the ICE's Engineering Solution Core Objective areas of "Defining the problem", "Designing a solution" and "Specification and measurement" and to their Professional and General Core Objective area of "Communication and reports".

(b) senior undergraduates

The MLE is also aimed at second and third year Civil Engineering undergraduates.

The syllabus for the BEng (Hons) Civil Engineering course at the University of Plymouth is detailed in the Definitive Course Document for BEng (Hons) Civil Engineering (September 1993) from the University of Plymouth's School of Civil and Structural Engineering.

The document states that the course aims to provide a technical education which develops a first Degree level of competence within the field of Civil Engineering. The objective is to give graduates the educational background necessary for the formation of Chartered Civil and Structural Engineers.

The course adopts an integrated approach to theoretical and practical teaching related to the needs of industry. Theoretical, practical, design and project work is integrated, to progressively greater degrees as the course proceeds. The course aims to provide students with a sound understanding of appropriate mathematical and numerical methods; surveying; engineering materials; engineering analysis; engineering design; and construction management and

methods. The integrating theme of engineering application throughout the course, emphasises the importance of industrial context, management and communication. The applications content increases with each year of the course.

The course also aims to develop an enquiring mind and the ability to extend personal knowledge as the need arises; to cultivate a systematic approach to the modelling of real problems; to give experience in experimental and numerical procedures; to develop an ability to communicate effectively in an engineering environment; and to develop a creative design ability. A balanced course of education is further achieved through the inclusion of a programme of field courses, modern languages, site visits and industrial counselling.

Civil Engineering is arguably the most wide ranging of all the branches of Engineering. Certainly, within a three year period it is impossible both to cover all aspects and to maintain the depth of treatment appropriate to an Honours Degree. The Course addresses this problem by covering the main scientific principles, together with their application to design and construction, in a focused set of compulsory core modules. Additional breadth and depth are provided in a limited number of other areas by optional course components.

A wide range of teaching strategies are adopted, including lectures, example classes, seminars, tutorials, group work and projects. Personal skills and creativity are developed through practical work, field courses and design projects. Communication skills in written, spoken and graphic forms are developed through a variety of exercises related to the Civil Engineering topics. The assignments are designed to ensure that the students spend time both working independently and working collaboratively in teams.

In the final year of the course the core disciplines continue to be developed but there is a greater emphasis on design and the application of theory within

selected subject areas. In the final year, demands are made of the students' personal and communication skills in ways which model those likely to be found in their future careers. Thus, experience of intensive team working is provided by the Specialist Design and Construction Project which must be completed in a three week period when most other activities are suspended, whereas the individual Project focuses on self-motivated research conducted throughout the final year. Both group and individual reports must be produced and defended at oral presentations.

The above description provides an overall feel for a typical Civil Engineering degree course and it can be seen that the skills and knowledge needed by a senior undergraduate are similar to those needed by a new graduate trainee. Details of some of the specific skills and knowledge that are needed are examined in 'Chapter 4. Knowledge acquisition'. Chapter four covers the elicitation, from 3 categories of expert, of an essential, representative set of skills and knowledge which it was felt the MLE could help to convey.

The material cover by the MLE can be seen to relate to the material concerning free surface flows covered in the University of Plymouth's BEng (Hons) Civil Engineering final year double module of Hydraulic Engineering III. This module extends the theoretical base established in Hydraulic Engineering I and II (taken in years one and two respectively) and applies the knowledge to commonly encountered practical problems associated with both free surface flows and confined flows.

3. Approaches to developing CBL: the need for a method

This chapter begins with a review of a range of existing products and approaches to the design of computer based learning, and discusses their strengths and weaknesses in the context of our chosen learning domain and target learners. The purpose of this review was to identify valid, and invalid, approaches which could be considered for the design of multimedia learning environments. An initial approach to the development of the MLE by means of simply combining a number of different styles of computer based learning, which proved inadequate, is briefly examined. This is followed by a review of some existing methods for the design of MLEs which provide a more systematic framework for developers. The chapter concludes with a description of the approach finally adopted for the present research, which draws to a large extent on the methods and principles of knowledge engineering used in the design of knowledge based systems. Such an approach was used because it is not driven by the need to try out a particular technology or learning theory, but led by (a) an understanding of the learners' knowledge needs (b) an analysis of the content of the required domain knowledge which guides (c) the choice of which learning mechanisms are appropriate and (d) the decisions over technical issues about the type of system to implement.

At the beginning of the research the developers considered a number of different styles of CBL system, these included: tutorial; drill and practice; intelligent tutoring; hypertext; simulation and microworlds. Each of these styles is reviewed in 'Section 3.1. Styles of CBL'.

It appeared to the developers that, over the years, a number of different teaching systems had been produced in response to shifts in psychological paradigms and advancements in delivery technology. These teaching systems could be

placed into the broad categories above in which all the programs within a particular category tend to be based on the same underlying learning paradigm and to have been developed using similar design methods.

The current level of sophistication in computer technology means that it is now possible to incorporate multimedia elements relatively easily into a program from any of the above traditions of computer based learning.

It seems from the literature that developers of new computer based teaching systems often tend to begin from the outset, either consciously or unconsciously, with a particular final style of learning system or particular model of learning in mind. This tends to predetermine both the types of knowledge which they will look for and incorporate into the system and the ways that they will consider to structure and sequence this knowledge.

Originally, this was the type of approach adopted by the developers. As discussed in 'Section 3.2. Initial ideas for the development of the MLE' it was intended to develop the MLE by simply combining a number of CBL styles, in particular hypertext and intelligent tutoring systems to form a kind of "Expertext" (Rada, R., (1991)). It was intended to develop a teaching strategy based on intuitions and insights from the experts involved, ad hoc, as the work progressed.

This "Expertext" approach would have resulted in the production of a complicated "brain-dump" which could immediately be seen as being a useful, rich and personal reference base for the expert who had helped develop it. However, it was difficult to produce the essential guidance mechanisms and overall teaching strategies that would be needed to allow a relative novice to become familiar with and learn from the material. The explosion of cross-references which the expert perceived would tend to obscure or bury the main routes through principal concepts needed by the learner.

It was realised, as overviewed in ‘Section 3.4. A method for the design of MLEs: the objectives of the research’, that a different approach would have to be adopted. The weakness of our initial approach was that it was based on a collection of *techniques* with insufficient attention to a coherent design *method*.

The intention of the research is to recognise that interactive multimedia needs to be treated as a new medium with its own design method, rather than just an expedient selection of techniques from earlier technologies. A number of recently proposed methods were reviewed but were felt to fall short of the goal of enabling designers to exploit fully the power of multimedia in ways which take advantage of appropriate mechanisms of human learning. Having tried and felt dissatisfied with existing methods and approaches, specific objectives were identified for an improved method.

First, a detailed understanding was needed of the types of knowledge present in the domain, bearing in mind the target learners and the context of their working and learning environment. Secondly, this understanding would help to determine how each identified knowledge type could be best understood by the learner using various media to take full advantage of the possibilities of interactive multimedia within such a rich domain.

Thirdly, as it had been realised that the developers intuitions alone produced an inadequate teaching strategy, the knowledge analysis should lead to a suitable teaching strategy or pedagogic model that would integrate all the knowledge types while still enabling them to be presented in their most appropriate manner.

In order to demonstrate the way in which these elements of a design method can be integrated, the whole MLE would be designed in detail. Key parts of it would be implemented and evaluated.

3.1. Styles of CBL

The general categories of CBL system described in this section, that is:

- “Drill and Practice”
- Tutorial
- Intelligent Tutoring Systems
- Hypertext/Hypermedia
- Simulation
- Microworlds

are widely discussed by many writers. However, the precise definition of a category sometimes varies between authors and the dividing line between certain categories is often unclear, for instance, it is debatable how wide ranging and complex a “Simulation” has to become before it can begin to be described as a “Microworld”. Also, occasionally, a particular system, such as the one developed through this research, may combine elements from a number of categories.

Some authors might not recognise some of the categories presented here, as separate categories in their own right. For example, “Intelligent Tutoring Systems” may be seen more as an underlying technology that is capable of implementing some of the other styles of CBL such as “Simulations”, “Microworlds” and “Tutorial” dialogues. If “learning by browsing” is not seen as a pedagogic strategy then “Hypertexts” might be deemed to be merely containers of information, together with electronic books, databases and electronic documents, which all need to be organised by some external pedagogic strategy to be useful to a learner.

Some authors may suggest additional categories. For instance, “Games”, which are not considered here as a separate category, as a system developed from within any of the given categories could be implemented in a game format.

General, commercially available computer tools or applications such as "empty" databases, spreadsheets and word processing applications are also not considered as separate categories. However, the guided "filling in" of named cells and experimentation with a spreadsheet that had already had the linking equations entered into it could be included, although, this use of a spreadsheet could be thought of as a kind of "Simulation". Also, the use of computers as a means for mediating communication between learners, as part of a process to help them learn, is not considered here.

The categories of CBL which are discussed here have been selected because (a) they cover the great majority of types of system described in the literature, (b) some of them were used as the basis for an initial prototype (discussed in 'Section 3.2. Initial ideas for the development of the MLE') and (c) the finally designed and implemented MLE can be seen to incorporate elements from several of the categories. This is demonstrated in the chapters on design and implementation.

The main purpose of the review is to identify the strengths and weaknesses of different CBL design approaches in order to evaluate their potential contribution to MLE design. A further reason for analysing previous approaches to CBL is to show that they would be inadequate on their own for the target learning needs. This discussion forms a platform from which our final design method develops.

The present section also helps illustrate how the general approach used by the developers (as discussed in 'Section 3.4. A method for the design of MLEs: the objectives of the research') does not incorporate any initial bias towards a particular final style of CBL implementation.

3.1.1. “Drill and practice”

Examples of recently available commercial "Drill and Practice" programs:

- **The Rosetta Stone Language Library** (Macintosh & Windows/MPC)
Fairfield Language Technologies, Harrisonburg VA, USA, 1994.
92 chapters of, CD-ROM based, carefully sequenced multimedia (pictures and speech) instruction for the new language learner (available in Spanish, English, French and German).
- **Tables Master** (Macintosh)
Flexible Learning Company, 1994.
A suite of programs to teach, consolidate and test the Times Table. A record of achievement is available and the program will even warn you when it thinks you need more practice.
- **Word Attack Plus!** (Macintosh)
Davidson, 1994.
A set of five different learning activities for students aged 9 to adult that help to build up vocabulary. The activities offer different ways of choosing and matching the correct word to a phrase or situation (9 different levels).
- **Yearn To Learn: Snoopy** (Macintosh)
Image Smith, 1994.
Activities include Maths, word, and music games. Arithmetic games include addition, subtraction, multiplication and division. The Word games test word recognition and spelling. As they get older just increase the skill level.

“Drill and Practice” programs offer a structured succession of increasingly more difficult testing exercises that are designed to help the learner gain some skill or knowledge or are aimed at guiding the learner towards the correct performance of a procedure. Questions might be written into the “Drill and Practice” package, drawn at random from a collection in an item bank or, as in some cases of simple numerical questioning, actually generated by the software.

The learning objectives of a “Drill and Practice” program are set by the program's designer. Each set of “Drill and Practice” exercises tend to be focussed on an individual activity. The learning objectives seem obvious to the learner and are seen as the level of performance of the given activity that they are expected to be able to accomplish.

A range of verbal, numerical and spatial activities can be effectively taught on a

computer by the “Drill and Practice” method. However, the performance of many activities requiring, for instance, detailed motor skills, non-verbal communication or complex social behaviour cannot be described adequately by any means that can be entered into and then evaluated by a computer. So, these activities have to be practiced by other means.

Given the appropriate activity, a computer can be an extremely effective means of providing learning through practice. The increasing familiarity with the knowledge or procedure that is produced from repeated practice enables the learner to perform in an ever more “automatic” manner. For instance, language learning, spelling, simple arithmetic and other basic skills can be usefully practised until they become automatic to the learner and then they can continue on to more complex skills or knowledge.

“Drill and Practice” programs need to be used with other forms of teaching because, individually, they do not provide background information and do not directly help in transferring the skills into new contexts, or in integrating skills and knowledge. Their strength lies in consolidating the fundamentals to allow the learner to then concentrate on higher level aspects. Because of their stress on producing automatic performance, “Drill and Practice” programs usually avoid tuition or intrinsic feedback (that is, sensory feedback occurring naturally as a result of an action and which tends to inherently provide information about how to improve, for example, bowling at cricket stumps). Instead, they typically give only basic right/wrong feedback, with no information about how to improve, over a large number of exercises.

In conclusion, modern “Drill and Practice” programs can be seen to be based on the much older concept of linear programming. The theory underlying linear programming was derived from the behaviourist principles of operant conditioning, the basic law of which states:

"if the occurrence of an operant is followed by the presentation of a reinforcing stimulus, the strength is increased"

(Skinner, 1938).

Within this theory teaching was:

"simply the arrangement of contingencies of reinforcement"

(Skinner, 1968).

The questions presented by a linear program were structured to cover the material in small steps progressing from the simple to the more complex. An effort was made to minimise the possibility of incorrect responses being made by the learner as they progressed. After answering a question the learner would be told whether they were right or wrong (their responses would usually be either completely right or completely wrong) but would receive no other feedback. The questions always appeared in a set sequence regardless the learner's answers. The only individualisation available to the learner was in the pace at which they worked through the questions.

More recent work in learning theory, arising from within the Constructivist school exemplified by researchers such as Vygotsky (1962), Bruner (1966), Piaget (1970) and Papert (1980) has emphasised that human learning is not simply a process of conditioned responses. Rather, they suggest that learning involves the learner in a personal exploration of the material in which they construct their own understanding by involvement and interaction.

3.1.2. Tutorial

Examples of recently available commercial tutorial programs:

- **AlgeBlaster Plus** (Macintosh)
Davidson, 1994.
AlgeBlaster Plus is an interactive tutorial designed to help pre-algebra students understand the algebraic process, learn the basic steps in solving equations, and practice graphing skills.
- **Algebra** (Macintosh)
Broderbund, 1994.
An interactive first year Algebra tutor with animated graphics and digitised sound. It covers the language and rules of Algebra and, among other aspects, Absolute Value, Polynomials, Linear Equations and multiplication and factoring.
- **Geometry** (Macintosh)
Broderbund, 1994.
Turns the Mac into a geometry teacher that takes you through the principles and concepts of geometry and explains them by bringing them to life visually and allowing you to manipulate them with the mouse.
- **Physics** (Macintosh)
Broderbund, 1994.
Covers the basics of Physics, serving as an extension of classwork, a refresher course or private tutor. Like Geometry, it explains the concepts and provides a visual environment in which to interactively apply the concepts as a way of understanding them.

Tutorial programs form the majority of CBL programs that are available, although, they do not appear to be the most frequently used form of CBL.

In tutorial programs, as the name suggests, the computer acts as a “tutor”. The intention is to provide computer based teaching whilst minimising the need for the presence of a human teacher. Subject matter is presented to the learner in the form of text, diagrams, pictures, animations, video, and so on. The learner then has to answer questions on the topic just covered by entering letters, words, numbers or by selecting areas of the screen. Feedback is typically provided by means of a mark indicating the number of correct answers, additional explanation of the subject matter, and, perhaps, a comment or exercise designed to specifically remediate the error made by the learner.

The information presented in a computer tutorial is generally quite basic because

it is presumed that the material has been covered elsewhere by the learner.

"Tutorials may be useful for review after initial teacher instruction, especially for those students who need to spend more time on the topic than the rest of the class. Such self-paced experiences could also be useful as a make-up mechanism for those who missed the teacher's original presentation, or as a substitute for human instruction where no teacher is available."

(Roblyer, 1988, page 10).

The learning objectives of a tutorial are set by the program's designer, and are usually given at the start of the tutorial together with a summary of the tutorial's content. Additionally, an appraisal of the prerequisite skills needed for the tutorial may be provided. This allows the learner to determine whether they are ready for the tutorial or, perhaps, first need to do other tutorials or to be given additional instruction.

The primary purpose of a tutorial program is to examine and correct the understanding learners have of what they have just learnt. Learner's usually interact with the tutorial by entering answers to questions.

Two methods of answering questions can be provided:

(i) multiple-choice

and, (ii) concealed multiple-choice (Laurillard, 1991a)

Basic multiple-choice questioning and answering is by far the most commonly used format within educational software. With this format each question is presented together with a list possible answers. For each question only one of the listed answers is correct and the others typically cover a range of plausibility. The learner chooses an answer and is given the appropriate feedback. The software will usually be able to provide a tailored response, such as a further explanation or hint (and then the asking of another question), for every one of the possible answers which could be selected. Even correct answers usually result

in further explanation because the learner may, in actual fact, have had no idea what the real answer was and might merely have made a lucky guess.

There are two drawbacks with this type of multiple-choice questioning. First, learners are not allowed to put forward their own ideas. Secondly, learners' misconceptions may be encouraged as they think about and select a plausible but incorrect answer. Although the correct answer will be given it will be difficult to erase the memory of the thinking that was done to believe in the wrong answer in the first place (Laurillard, 1991a).

Concealed multiple-choice questioning goes some way towards overcoming these drawbacks. With this format the same list of answers is stored but not displayed. The software attempts to match the word or phrase entered by the learner to the stored answers. As the learners do not see the stored answers they are not deceived into contemplating and making incorrect selections. Moreover, allowing learners to think up their own answer should help them to reflect upon and integrate with their previous knowledge to a greater extent than if they merely had to select an option. Also, an additional advantage is that many more possible answers than could reasonably be displayed can be stored to be matched against learner input. However, this format is more difficult to program and, of course, the designer will still not be able to cover all the possible answers which might be given.

The effectiveness of a tutorial program can be measured by the appropriateness of the feedback it provides. The simplest and most prevalent type of tutorial program will only discriminate between correct and incorrect answers. Better programs will differentiate between different kinds of mistake and will offer more finely tailored and therefore more meaningful feedback. This should enable the learner to better integrate the new knowledge as they will be helped to recognise how they went wrong.

In conclusion, tutorial programs clearly do not conform to the strict principles of operant conditioning and linear programming discussed in 'Section 3.1.1. Drill and Practice'. Skinner, in his theory of operant conditioning, deliberately avoided any consideration of the internal "cognitive" processes of the learner. He preferred to deal with measurable behaviour.

Tutorial programs, however, do attempt to consider and correct the thinking done by the learner whenever they answer a question. Multiple-choice questions are favoured because answers can usually be more or less acceptable, rather than the totally correct or totally incorrect answers typical of linear programs. The units of information presented also tend to be larger in a tutorial program as the designer does not have to ensure the learner always responds correctly. Tutorial programs also "individualise" learning not merely by allowing learners to work at their own pace but by presenting different learners, because of their differing answers, with very different routes through the material. In this way they can be considered to be somewhat Constructivist rather than strictly Behaviourist in approach.

The basic format of a tutorial program can be seen in Crowder's "intrinsic programming" approach:

"the essential problem is that of controlling a communication process by the use of feedback. The student's response serves primarily as a means of determining whether the communication process has been effective and at the same time allows appropriate corrective action to be taken."

(Crowder, 1959)

3.1.3. Intelligent tutoring systems

Examples of recently available, commercial intelligent tutoring systems:

- **APT (The Application Program Tutor)** (PC)
The Electric Brain Company, Leeds, UK, 1994.
APT has knowledge about interfaces which enables courses to be constructed quickly. The author demonstrates to APT the activities the learner should master; constructs a hierarchical task analysis of the activities; and specifies which concepts the learner should understand and explains how they relate to the procedures. There are several teaching strategies, and the teaching varies both in the subject matter covered and the teaching strategies used based upon a model of the individual.
- **ECAL**
The Electric Brain Company, Leeds, UK, 1994.
ECAL provides guided access to large collections of multimedia information. The author creates a number of multimedia fragments (both presentation and diagnostic) and indexes these according to the concepts which they teach or test. From this information, ECAL derives a concept map of the available material and assigns an importance and generality to each concept as well as identifying clusters of related concepts. ECAL directs the learner through the appropriate parts of the material maintaining a smooth focus of interaction and moving from topic to topic as it acquires evidence that the current topic is sufficiently understood.

Intelligent tutoring systems (ITSs) attempt to make the computer act as "tutor" by using techniques from the field of Artificial Intelligence. ITSs may be thought of as attempts to produce in a computer behaviour which, if performed by a human, would be described as "good teaching" (Elsom-Cook, 1987).

Although there is a great deal of variation in the structure of individual ITSs, many researchers in the field (for instance, Wenger, 1987; Burns & Capps, 1988; Mandl & Lesgold, 1988) typically expect to encounter four major elements in an ITS:

- I. the *expert knowledge* component
- II. the *student modelling* component
- III. the *tutoring* component
- IV. the *user interface* component

I. The *expert knowledge component* contains the domain's facts and rules which are to be conveyed to the learner. It serves two purposes. First, it provides the domain knowledge which includes the generation of questions, explanations and responses. Secondly, it offers a standard against which the learner's performance can be judged. To judge the learner's work it must be able to generate problem solutions in the same way as the learner, so that both solutions can be compared. If the learner is to be monitored as they solve a problem then the expert component must be able to generate several realistic solution paths to the problem to ensure that every step of the learner's work can be correctly evaluated.

Obviously, it is important for the expert knowledge component to use a psychologically valid and accurate model of human performance. However, in practice, this can be difficult to achieve. For instance, Clancey (1984) warns against the use of off-the-shelf expert systems because they tend to employ computationally tractable algorithms rather than cognitively viable algorithms that can be communicated to the learner. Also, only incorporating one model for solving the problem may in itself be naive as Ridgeway (1991) suggests that a major characteristic of an expert is that they can solve the same problem in a number of ways. Experts can also, as they solve a problem, translate between different representations and relate and reconcile different approaches and modes of representation.

The expert knowledge component obviously only embodies the designer's particular view of the domain. This can be problematic if the learner cannot understand the ITS's explanations or if the ITS is completely unable to reconcile its solution with the learner's (Wenger, 1987). Naturally, human teachers have their own viewpoint but they are able to adjust it to allow them to successfully relate to the learner.

II. The *student model* is used to dynamically record the learner's emerging skills and knowledge. The majority of researchers in ITSs tend to believe that an ITS should possess a student model (Hartley & Sleeman, 1973; Self, 1974, 1987, 1988; Rich, 1979; Sleeman, 1985; Tobias, 1985; Zissos & Witten, 1985; Ross et al., 1987; Gilmore & Self, 1988; Wachsmuth, 1988). However, providing a detailed and wide-ranging student model is a difficult if not impossible task because the method of entering information into the computer, usually a keyboard and a mouse, is so limited.

"Understanding the processes that someone else goes through as they solve problems is hard for a human observer... One of the main problems which human observers face is that they are presented with a multiple parallel stream of cues about internal processes such as: diagrams; writing - odd notes, doodles; speech - often in fragmentary form; odd analogies and slang expressions; and a range of gestures such as pointing to different parts of the figure or gazing towards particular pieces of written work; social cues which yield evidence of understanding, ignorance, frustration, withdrawal, and the like. Each of these cues can have some relevance to the problem-solving process. None of them is accessible to the computer."

(Ridgeway, 1991, page 131)

The information contained in the student model is usually used by the ITS's tutoring component to help guide the system's pedagogic interactions with the learner. A number of detailed approaches to student modelling have been developed. There are two popular, broad traditions in student modelling which are commonly referred to as "buggy" modelling (Brown and Burton, 1978) and "overlay" modelling (Goldstein, 1982).

In "buggy" modelling, the student model represents what are believed to be the current errors that are being made by the learner or what are believed to be the current misconceptions that are held by the learner. However, Ridgeway (1991), suggests that for even simple domains there can arise a large number of naive models which can be difficult to diagnose. Learners may exhibit a variety of idiosyncratic methods and approaches - many of

which may work quite effectively! And, as the domain becomes increasingly more complex, so, the range of possible naive models also increases dramatically. It appears likely that no complete set of naive models could ever be produced for a complex domain. In any case, Ridgeway (1991) doubts the value of such an approach:

“At a more principled level, if we take the view that the machine is there to explore the user’s naive model and to provide some diagnoses and subsequent remediation, then we are assigning an important learning task to the machine which could be performed by either a fellow pupil or by a teacher. From the learner’s viewpoint this may be of no consequence. However, from the fellow pupil’s or the teacher’s viewpoint, it may have undesirable consequences in that they will thereby lose the opportunity to engage in a form of learning situation of considerable complexity and potential value. It seems a rather strange proposition that one should write computer programs which impoverish the potential for learning by teachers or by other pupils.”

(Ridgeway, 1991, page 134)

Anderson (1993) also questions the educational validity of bug diagnosis. He suggests that research in a number of domains (e.g. Anderson and Jefferies, 1985; Payne and Squibb, 1990) has demonstrated that most learner errors are not caused through misconceptions. Most errors are often just slips. The remainder are mainly caused by learners’ guessing what to do next when they come across a new problem where they simply don’t know what to do. So, in these situations, it is more a case of no knowledge as opposed to defective knowledge. Also, there is not much evidence which proves the effectiveness of feedback on misconceptions. Anderson (1991) cites the most notable research on this subject as being conducted by Sleeman, Kelly, Martinak, Word and Moore (1989) who found that explaining errors to students learning algebra produced no benefit. However, maximal benefit was produced from simple reinstruction on correct conceptualisations and procedures.

In “overlay” modelling, the other general, popular approach to student modelling, a record is maintained of the learner’s grasp of each component of

knowledge that is represented in the expert module. So, at any point in time the learner's state of acquired knowledge can be viewed as a subset of the expert's.

The idea of decomposing the knowledge that is to be taught into smaller units has a long history in educational psychology (Gagné and Briggs, 1979; Thorndike, 1922). However, this type of approach has been challenged by some. For instance, structuring the teaching of a cognitive skill by means of a pre-specified student model may seem to conflict with the approach put forward by constructivists. Constructivists would argue that a student learns a new skill by "constructing" their own individual understanding of it rather than by just adopting a given external specification.

ITSs using an "overlay" modelling approach have also been criticised because of the constraints they can place on student behaviour (e.g. Bonar and Cunningham, 1988). One, sequential, constraint which has been suggested concerns immediate feedback. The ITS will typically provide immediate feedback to ensure each step of the learner's solution matches the expert's solution. So, not only does the student's eventual solution map onto the expert's, but each successive step maps onto the expert's.

A number of criticisms of immediate feedback have been made. These include, first, learners are deprived of the chance to realise for themselves that they have made an error. Secondly, immediate feedback may prevent learners from developing their own meta-cognitive, self-monitoring skills (Chi et al, 1989; Schoenfeld, 1985).

The research carried out by Corbett and Anderson (1990) and Anderson (1993) into feedback control in programming tutors showed that allowing students to discover, debug and correct their own errors only slowed them

down and did not lead to improved learning. Although, this issue of when to provide feedback in various circumstances is still very much open to debate and further research.

“Overlay” modelling, however, does not necessarily have to result in the provision of immediate feedback. The only requirement of “overlay” modelling is that the learner’s solution is, at some point in time, matched against the expert’s solution. Anderson (1993) has experimented with demand feedback (feedback is only given when the learner asks for it) and flag tutoring (errors are pointed out to the learner when they are made but the learner can ignore the ITS and carry on as they were). However, both these modes of tutoring proved inferior to immediate feedback.

It has also been argued that “overlay” modelling tutors can teach learners meta-cognitive, self-monitoring, problem solving skills, providing these are some of the skills which the programs have been specifically designed to teach (Anderson, 1993).

- III. The *tutoring component*, sometimes known as the teaching strategy or pedagogic module, is the part of the ITS that constructs and controls the teaching interactions with the learner. This module is closely connected to the student model and uses knowledge about the learner, together with its own pedagogic goal structure, to determine which teaching activities to present next, for instance: hints to overcome impasses in performance, advice, support, explanations, different practice tasks, tests to confirm hypotheses in the student model, and so on (Self, 1988).

The sequence and presentation style used for each topic can greatly affect the student’s learning experience. As discussed, when examining the student modelling component, the immediacy of feedback can be varied. As such, the

tutoring that is provided by ITSs can be arranged along a scale ranging from those systems that monitor the student's every move, continually adjusting their behaviour in accordance with the learner's responses and never relinquishing control, to *discovery* learning systems where the learner has complete control over the activity, and the only way the ITS can influence the activity is by adjusting the environment. In the centre of the scale are *mixed-initiative* systems where control is shared between the learner and the ITS as they exchange questions and answers in a kind of Socratic dialogue.

Many ITSs designed to promote discovery or exploratory learning are built around a "Simulation" or "Microworld" in such a way that intelligence can be incorporated into both the simulation and the pedagogic component. Some systems allow learners to interact without providing any guidance. The paradigm for this is the Steamer system (Hollan et al., 1984), which simulated a power plant. However, it is more usual for systems to offer some assistance or a kind of *guided-discovery* learning. An early example of this is Sophie (Brown et al, 1982) which taught electronic circuit troubleshooting by using a simple simulation of a conventional circuit together with intelligent comments on the learner's behaviour as they worked (for a fuller discussion of simulations and microworlds please see 'Section 3.1.5. Simulations' and 'Section 3.1.6. Microworlds').

IV. The *user interface* component mediates between the learner and the ITS. It is crucially important to the effectiveness and acceptance of the ITS and is therefore usually acknowledged as a vital element of any ITS. Progress in multimedia technology has delivered increasingly more sophisticated methods of computer-based communication which can increase the effectiveness of the learning experience. In addition, the ease of use and attractiveness of the user interface can prove essential to the learner's acceptance of the ITS. Current ITSs provide user interfaces which can use for

input, for example, mouse clicking at certain screen locations, selecting and dragging objects by mouse, menus and a fairly free treatment of pseudo-natural language (both spoken and typed). ITS output can range, for example, from merely displaying pre-stored “canned” texts typical of “Drill and practice” programs, to the use of fairly complicated generic frames containing sophisticated multimedia elements.

In conclusion, ITSs can be seen as an attempt to improve upon the earlier CBL styles of “drill and practice” and tutorial programs by offering increased flexibility in dealing with both the learner’s input and in providing more suitable output. Instead of pre-determined content, ITSs can generate new material. Rather than having pre-determined feedback with limited, fixed routes through the material, an ITS can be more adaptable and responsive in its teaching.

Research into ITSs has produced a number of interesting techniques that can be usefully applied to the development of many CBL systems. However, there are some intractable problems which makes the idea of providing an ITS which could replace a human tutor, in even the most appropriate and restricted of areas, an extremely difficult, time consuming and, some would say, misguided endeavour. For instance, teachers and fellow pupils can be very good at, and can themselves learn a great deal from, the process of diagnosing and helping an individual to overcome their shortcomings. Clearly, depending on the context, some teaching tasks can more appropriately be allocated to the trainer, fellow pupil or individual learner, than to the CBL package.

In terms of educational effectiveness, many ITSs have been criticised because they are not based on any coherent, scientific theories of teaching. This has been rectified, to some extent, by later work in the field which has been conducted by cognitive scientists who have brought with them theories of learning and instruction. However, much of the early work into ITSs was done by researchers

in artificial intelligence (AI) who were attracted to the field because they realised that any attempt to encapsulate the intelligence of the instructor would involve a number of the more difficult and important areas of AI research.

“Early work was driven by the challenge of bringing artificial intelligence techniques to bear on education, but often lacked a coherent, scientific theory of effective education.”

(Anderson, 1993, page 242)

AI researchers tended to develop teaching interventions by intuition and to measure the success of a system not by how well it instructed, but by how well it handled some of the difficult problems of AI.

3.1.4. Hypertext/Hypermedia

Examples of recently available commercial products or systems where the content material is structured and accessed using hypertext/hypermedia techniques:

- **Microsoft Dinosaurs** (Windows & Macintosh, Version 1.0)
Microsoft Corp., Seattle, Wash., USA, 1993.
(CD-ROM engine by Cognitive Applications Limited, Brighton, UK, 1993.)
A single CD-ROM of browsable, linked multimedia reference material on Dinosaurs. Articles and media elements are listed under a number of headings: Atlas, Timeline, Families, Index, Dinosaur Movies and Guided Tours. Clicking on highlighted text within an entry brings up a further explanation and a button which can be used to open a related entry. Buttons are always available to allow the reader to backtrack through the entries they have seen or to return to the main menu.
- **Microcosm** (Currently available for: DOS/Windows 3.1 and Windows emulation under OS/2 & Windows NT. Under development, as of Dec '94, for: MAC & Unix (running X-Windows) platforms).
Microcosm Team, University of Southampton, ongoing development.
Microcosm is an open hypermedia system which allows users to browse and query large collections of multimedia material that may have been produced using a variety of third party applications. Microcosm can be seen as an 'umbrella' environment allowing the author to make links from documents in one application to documents in another. All data files remain in the native format of the application that created them. All link information is held separately in link databases (linkbases). This allows documents held on read only media such as CD-ROM and videodisc to be integrated and makes it possible to have different sets of links for the same data, so individual readers can produce their own customised links. A spectrum of link possibilities exists, from those explicitly defined by the author to those dynamically generated by the computer. A history mechanism maintains a list of all documents that have been visited and allows the user to return to a particular document. A mimic facility allows the user to follow a pre-defined tour through the documents. The user can also keep track of important items by using Microcosm's bookmarks. Microcosm can search data for related items based on a keyword or phrase, and rank them for relevance. Microcosm has been used within the educational field, for large archives, an Urban Information System, for computer aided engineering and for delivering multimedia materials in a range of other applications.
- **The World Wide Web on the Internet**
The Internet is an open world-wide communications network, linking together countless thousands of computer networks, through a mixture of private and public telephone lines. Its component networks are individually run by government agencies, universities, and commercial and voluntary organisations. No single organisation owns or controls the Internet, though there is an Internet Society that co-ordinates and sets standards for its use. Individuals, who are not part of a linked institution, can gain access by using a commercial service provider. The World Wide Web consists of several million "pages" of information, stored on host

computers throughout the world. Text, graphics, video clips, sounds and - most importantly - hypertext links to other pages are encoded into the World Wide Web's page format by the use of a language called HTML (HyperText Markup Language). These pages can then be displayed on a computer by using a "Web Browser" - a program which can understand and decode HTML. Clicking on a page's hypertext link calls up the related page, which may be in the same computer, or in another machine on the other side of the World. To help find information on the Web there are several directories which provide structured entries into the mass of pages. There are also a number of search engines or "Web Crawlers", that will track down specific topics. A Web browser can also allow access to most Internet newsgroups and sites where files can be downloaded.

One early definition of hypertext was given by Nelson in 1974:

"By 'hypertext' I mean non-sequential writing. ... Writers do better if they don't have to *write* in sequence (but may create multiple structures, branches and alternatives), and readers do better if they don't have to *read* in sequence, but may establish impressions, jump around, and try different pathways until they find the ones they want to study most closely. ..."

(Nelson, 1974, p. 85)

A *hypertext* system can be thought of as a collection of linked information nodes. When the information in the nodes contains more than just text, for instance, if graphics, video, sound and so on are included, then the system is often referred to, more generally, as a *hypermedia* system. Although some authors (like myself) use both terms interchangeably while others still prefer using the traditional term *hypertext* for all systems (Nielsen, 1995).

Hypertext was envisioned as far back as 1945 by Vannevar Bush (Bush, 1945) in his proposal for a system called Memex ("Memory Extender").

"A device in which an individual stores his books, records, and communications, and which is mechanised so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."

The actual term "hypertext" was coined about twenty years later by Ted Nelson as part of his audacious Xanadu project to develop a single repository of

everything that had ever been written (Nelson, 1987).

The nodes in a hypermedia system, which represent the system's basic unit of information, can vary in size, content and modality. For instance, an individual node might appear as a single concept or illustration or as an entire article (McAleese, 1990).

Nodes are connected together by associative links and hypermedia systems are frequently said to mimic the associative properties of the mind (Fiderio, 1988). In fact, Bush (1945) assumed that:

“The human mind... operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by association of thoughts, in accordance with some intricate web of trails carried by cells of the brain.”

(Bush, 1945, p. 106)

Many cognitive theories concerning the acquisition of knowledge support Bush's idea of an associative network. In Schema theory (Rumelhart & Ortony, 1977), mental constructs are known as schemata. Each schema that an individual constructs represents a miniframe in which they interrelate elements and attributes of an idea (Norman, Gentner & Stevens, 1976). Schemata are associative structures of attributes and relationships. Hypermedia systems can be seen to model schema theory, with each node comprising a schema that is associated (linked) with other nodes in an associative structure. Active structural networks can also be seen as a theoretical basis for hypermedia. An active structural network is a framework of nodes with the nodes having ordered, labelled relationships connecting them (Norman, Gentner & Stevens, 1976). Nodes represent propositions and links express relationships. Active structural networks are used to represent what a learner knows. Similarly, hypermedia can be used to explicitly represent the knowledge of an expert (Jonassen, 1990; McAleese, 1990).

The links within a hypermedia system allow the reader to navigate the material non-sequentially to explore related information. The destination of a link can be either a part of a node or an entire node. A piece of information can form the anchor for multiple links, and similarly may be the destination of a number of links. Within hypermedia there are two basic types of links, referential and organisational (Conklin, 1987). Referential links simply connect two related nodes whereas organisational links convey the type of relationships that exists between nodes. Hypermedia material can be organised in a number of ways. The organisational structure of the nodes and links might correspond to the characteristics of the task the hypermedia system is to support or teach; basic relationships in the content material; or to the author's semantic structure.

However, a hypermedia author's semantic organisational structure and referential links may seem arbitrary to users of the hypermedia because each individual's knowledge structure is unique, based upon their individual knowledge and experiences. Similarly, the way each individual chooses to access and interrelate information is also unique. The success of a hypermedia system can depend upon the control it can give to learners to enable them to build their understanding of a particular knowledge domain according to their own individual needs. To do this successfully will rely upon the extent to which the possible routes through the hypermedia:

(i) *Match the cognitive process of the user's learning.*

Many cognitive theories consider learning to involve the re-organisation of the learner's cognitive structure. By modelling an expert's knowledge structure in the hypermedia system, a useful knowledge structure may be mapped more directly onto the learner's cognitive structure. A hypermedia structure may reflect the knowledge structure of an expert. The way an expert thinks could be explicitly modeled for the learner (Jonassen, 1987). Research has shown that the teacher's knowledge structure is mapped onto

the learner's knowledge structure over time and with instruction (Shavelson, 1972; Thro, 1974).

(ii) *Avoid disorientation (i.e. getting lost) within the subject matter.*

For instance, a major shortcoming in many hypermedia systems is their failure to provide adequate information about where readers are in the system. Readers of paper books can always tell if they are 'at the end of the book' or 'three-quarters of the way through it'. However, because hypermedia is non-sequentially organized and the 'middle' for one reader might be the 'end' for another, a reader can follow link after link and feel disoriented (Yankelovich, Meyrowitz, & van Dam, 1991). Many methods to orientate the reader have been attempted, including: illustrations of the information web; maps indicating all possible path options at a given time; and diagrams of the paths a reader has already taken. For example, MIT's Spatial Database-Management System displays a 'world view', an overview of Dataland with a 'you-are-here' marker (Bolt, 1979). However, some kinds of maps are easier to produce than others. For example, it is possible to create a map of a reader's path through the hypermedia material or generate a diagram showing all the possible links from the reader's current position, but as the number of connections and quantity of information increases, so does the difficulty of providing useful and readable maps of the entire information web. Also, authors may make circular references which can cause even more complexities in trying to graphically represent the web of connections. Since most readers cannot readily understand a diagram with hundreds of crisscrossing interconnections, the problem of distilling or summarizing the information must be addressed.

However, some authors suggest that simply modelling an expert's knowledge structure in a hypermedia system will only result in any novice browsing that material becoming quickly disoriented. For instance, Whalley (1990) argues that

hypermedia which is explicitly, visibly arranged according to an expert's knowledge structure can often fail to provide a sense of context for the individual novice browser. The context in such a knowledge base is the structure of ideas in memory, which is an important aspect of what many people believe distinguishes an expert from a novice. So, in a large system, another expert may be able to understand and navigate the many links as they would already have a similar structure in their own mind. However, a novice might simply become overwhelmed, unable to orientate themselves in such a strange new landscape with the result that they merely browse aimlessly and uncomprehendingly.

In addition, Hammond (1991) worries that the use of a hypermedia system does not, in itself, guarantee learning. He suggests that the freedom of movement available in a hypermedia system is not necessarily a sound basis for learning as the browsing is often passive rather than active. For successful learning, steps clearly need to be taken to involve the user to ensure that they are actively interacting with the hypermedia system.

One way of involving the user might be to require them to actively use the hypermedia for a particular purpose, for instance, to answer a question or to find supporting background material to provide guidance throughout an extended problem solving exercise or case study. In a longer problem the repeated return to the knowledge base for help in solving different aspects or stages of the problem might also help the user to not only understand an individual segment of knowledge but to understand a number of segments and to appreciate the various links that exist, in the hypermedia, between these segments.

Another way to involve the user might be by providing them with facilities to manipulate the hypermedia material. For example, a user might be able to make their own links between elements in the hypermedia or to add their own pieces of hypermedia information. In the extreme, a user may simply be given the tools to

construct, from scratch, their own hypermedia system.

The notion of the usefulness of providing facilities for manipulating the information in a hypermedia system seems to relate to Mayes (1991) idea of a cognitive tool which is basically any device, or technique, for focussing a learner's analytical processes which leads to the active and durable learning of the information manipulated or organised.

The methods outlined above, such as, giving browsers work to do where they have to consult hypermedia for answers or by providing facilities for them to make their own links in existing hypermedia or to annotate or add their own hypermedia material also go some way towards addressing Whalley's (1990) other criticism of explicitly expert structured hypermedia, namely, their lack of relevance to a novice.

As has already been noted, hypermedia can be used in three ways: as read-only information, as material that can be partly modified by the user or as a system for users to create their own hypermedia from scratch.

A myriad of different applications have been found for hypertext and hypermedia (Nielsen, 1995). The range of existing hypertext and hypermedia systems can be categorised in various ways, for instance, Conklin (1987) distinguishes between four classes of hypertext systems: browsing systems, problem exploration tools, macro-literary systems, and general purpose hypertext, while Rada (1991) identifies three standard categories of hypertext systems: small-volume ("microtext"), large-volume ("macrotext") & collaborative ("grouptext"). Rada's three categories are discussed below, in a slightly adjusted form, in an attempt to convey a feel for the general types of (sometimes overlapping) hypermedia systems that it is possible to develop. The categories below are also referred to again in the discussion in 'Section 3.2. Initial ideas for the development of the

MLE'. This section includes an examination of the styles of hypertext/hypermedia which influenced the development of the MLE.

- (i) **Small-volume hypermedia:** *Typically, a single application or document intended to be used by one person at a time.*

These types of system can include CD-ROM reference material (for example, the Microsoft Dinosaurs CD-ROM overviewed at the beginning of this chapter) and online documentation and help systems. Many current computer software packages (e.g., Microsoft Word 6.0, Microsoft Visual Basic 4.0, Macromedia Director 5.0 & Allegiant SuperCard 2.5) include extensive hypertext help facilities.

Many hypertext CD-ROMs and help facilities, although not allowing the user to create their own links or incorporate their own hypermedia material, do allow users to create "bookmarks" (direct, named links to a particular section, which will often then appear in a pull-down menu, for example, SuperCard 2.5's help facility) and "annotations" (user typed in text attached to sections, for example, Director 4.0's help facility). Some systems also allow their text to be copied into an included notebook or another application (for example, The New Grolier Multimedia Encyclopedia, Release 6).

Other systems may be designed to allow the user to create their own links or, because of the delivery platform, are able to be altered relatively easily by experienced users. For example, HyperCard stacks can be altered by an experienced user in possession of the HyperCard application program which was bundled free with every Macintosh sold by Apple from 1987 to 1992.

A number of applications allow individual users to create their own hypertexts from scratch on their own personal computer, including: HyperCard and SuperCard (both are currently only available on the Macintosh but both are planned to be available on PCs), Plus (both Mac and PC under Microsoft Windows and OS/2), Guide (recent versions restricted to the Windows platform) and ToolBook (PC).

- (ii) **Large-volume hypermedia:** *Typically, a system linking together a number (could be thousands) of documents (perhaps in different formats from different applications). These systems usually incorporate search and/or filtering mechanisms and often operate over a network.*

Microcosm (Davis et al., 1992, 1994; Hall & Davis, 1994; Hill & Hall, 1994; Hall & Woolf, 1995) and the World Wide Web (Berners-Lee et al., 1994) can be thought of as examples of this type of hypermedia system. Such “Macro-literary” systems do not represent a single document or application, but rather a collection of materials that are linked together by hypertext. These systems link documents and enable users to identify and access those documents. Using this approach many sophisticated instructional environments have been developed. For example, Intermedia (an authoring system developed in the mid-1980s at Brown University’s Institute for Research in Information and Scholarship and run over a network of workstations, for more details, see Yankelovich, Haan, Meyrowitz & Drucker, 1988), enabled learners in college courses to browse through interconnected knowledge bases in lieu of text-books.

On the World Wide Web links are embedded within documents and both links and documents have to be encoded in HTML. However, Microcosm uses a different approach. Microcosm allows documents to remain in their original format and it stores all links separately from documents in a

"linkbase". Linkbases can be managed by standard database management systems and documents remain compatible with the original applications which created them. In certain circumstances (depending on how "Microcosm aware" an application is), Microcosm is able to automatically keep track of any changes made to various links when a document is being manipulated in its native application and Microcosm can send messages to applications to instruct them, for example, to open a certain document and scroll it to a certain location. Having the links held separately allows different linkbases to be used with the same documents. For instance, a child using a particular linkbase to study the life of the Indian nationalist leader Mahatma Gandhi would be guided along a different set of options to a trained historian who was using a different linkbase to explore parts of Gandhi's life in detail (Peltu, 1996).

In a very limited way the World Wide Web can recently be seen to be moving closer to the Microcosm paradigm. Web pages and their links still have to be encoded in HTML. However, HTML Web pages can now refer to some documents and programs which remain in their native format and some native applications now have the ability to embed live Web links into the documents they are used to create. For example, Microsoft has released "viewers" (programs which can be used by Web browsers to display documents in their native format) for all their propriety applications. For the next version of Microsoft's office applications suite, Office 97, Microsoft has rewritten each of the main applications to integrate with the Web. For example, it is easy to embed live Web links into Excel spreadsheets and Word documents, the presentation package PowerPoint can be used to create animated Web pages and the database Access is able to update Web pages dynamically: change the Access data and the related Web page will also update. The collection of OpenDoc components making up Apple's Cyberdog browser allow live Web links to be embedded

in documents created by OpenDoc-enabled applications. Other developers are working with Microsoft's ActiveX component technology to produce "add-ons" for Microsoft's Internet Explorer 3.0 Web browser. Similarly, "plug-ins" have been developed that enable a Web browser to execute certain types of programs. For example, the Java plug-in allows Java applets to be executed, the "Shockwave" plug-in enables Macromedia Director applications to play and the "Roadster" plug-in permits the running of SuperCard programs.

As these types of hypermedia system can hold so much information they run the risk of overwhelming the user with too much material. To help overcome this such systems often incorporate a number of searching and "filtering" mechanisms. For example, Intermedia's "full-text" searching system allowed users to search the entire Intermedia database to find every occurrence of the specified text in all documents, regardless of type. With "filtering", the same criteria that can be used for searching can also be used by an author to restrict access to documents, specific readers or to groups of readers. For example, authors may ascribe differing levels of security to each of their documents. Alternately, some readers may wish to have a "filtered" view of the web of hypermedia information. For example, the extensive *Dickens Web* (Landow & Kahn, 1992) developed at Brown University can provide students with specific, filtered overviews of the hypermedia web. For instance, with particular reference to understanding the novel *Great Expectations*, an overview diagram can be produced of all the relevant authors about which information is held in the web that have influenced Dickens or that have been influenced by Dickens.

Filtering can also be done automatically and continuously in cases where the user wants to be kept informed about certain events. Good examples of this include the work that has been done on "personalised newspapers".

GMD in Germany (Haake et al., 1994) have worked on an experimental individualised electronic newspaper (IEN) that allows readers to see different views of a database of news items depending on their interests. The newspaper interface, designed by Klaus Reichenberger, can automatically lay out the current stories that match the user's stated interests in an interesting and appealing display. The newspaper is also interlinked with other hypertext services such as an online dictionary that can be used to look up words which the user does not understand.

In the Microcosm framework, holding link information separately from documents allows for the development of intelligent agents (Hall, 1996) which can filter information to match the needs of each user. The agents can be simple programs or more complex neural networks or expert systems (for example, Crowder et al, 1995), for a discussion about linking expert system techniques and hypermedia see 'Section 3.2. Initial ideas for the development of the MLE'.

(iii) Collaborative hypermedia: *systems that are created or accessed by a group of people.*

These types of system usually involve supporting a few related documents for a few collaborating users. Examples of this include hypermedia based CASE (Computer-Aided Software Engineering) tools (Bigelow, 1988). For example, *HyperCASE* (Cybulski & Reed, 1992) which supports software developers in project management, systems analysis, design and coding. It offers a customisable, integrated visual environment with tools for producing text and diagrammatic presentations. It enables software engineers to see different representations (i.e. the description of a process in the system requirements specification; the process in a dataflow diagram; the code associated with the process and so on) in parallel.

However, large-volume hypermedia can also be worked on collaboratively. For instance, one developed at Brown for English literature students included over 1000 documents and 1300 links within a term (Landow). It was used as the focus from lectures and enabled students to work together collaboratively, reading, extending, criticizing and revising the corpus of knowledge and judgements that were entered onto the system. It allowed changes to be made quickly and by incorporating student assignments it helped reduce the distinction in relative status between the academic staff and student information providers. Similarly, Microcosm has been used as a large-volume collaborative hypermedia system (for example, in computer aided engineering). The World Wide Web can also be thought of as a large-volume collaborative hypertext (it was originally developed at CERN as a small collaborative hypertext to facilitate electronic collaboration amongst scientists).

To some extent, it is even possible to work collaboratively on small-volume, stand-alone hypermedia. This could be done by either a group reading or revising the information together, or by individuals working sequentially on the hypermedia over a period of time.

A number of problems can occur when several users work together on a shared hypertext. The already discussed problem of becoming lost or disoriented may be made much worse for an individual user when the hypermedia is continually changing "behind their back" due to the activities of other users. One proposed solution to this problem, employed in a collaborative use of the NoteCards (Halasz et al 1987) authoring system (Trigg et al. 1986), was to establish a special area of the hypertext for communication among authors and to use a different typeface for the text written by each author. It has been suggested that, in general, hypertext systems could keep the identity of authors as attributes of the associated

nodes or links and then use this information when allocating change and deletion privileges (Nielsen, 1995). A typical example of this would be the use of Intermedia for teaching where the lecturer would be allowed to change, add and delete the hypertext material whereas the students would be authorised only to add links and annotations. Similarly, users could be given facilities to be able to "filter" their access to the hypertext according to author IDs.

Another problem made worse by cooperative authoring is that of version control (Delisle & Schwartz, 1987). For instance, say there are two nodes, A and B, that are linked together. Suppose an author revises node B and splits it into two new, separate nodes. The link from node A now needs to be re-directed to the most appropriate of the two new nodes. The hypermedia system should ensure this re-direction is done by the author as there is no easy method for automatically judging which of the new nodes would be the most appropriate.

The large volume and collaborative hypermedia approaches, such as those supported by Microcosm, which allow a tutor to mediate learner access to a wide variety of material for a protracted course of study (say, 1 semester) have many advantages. They will be considered later in 'Chapter 9. Design of the MLE' as a possible overarching mechanism for the linking of separate MLEs each covering single topics alongside other learning material within a subject.

Rada (1991) also considers the possibilities of a fourth category of hypertext: intelligent hypertext or "expertext". This category is considered in detail in 'Section 3.2. Initial ideas for the development of the MLE'. This section discusses one of the initial ideas of combining a hypermedia reference base with intelligent tutoring system techniques to be able to filter the hypermedia material to produce a "map" through the reference base indicating only relevant material for the set

enquiry with the given user and task context.

In summary, although using an associative network of linked multimedia nodes would intuitively seem to offer the possibility of providing a rich and effective learning experience there are a number of problems to be addressed, including:

- Difficulties in navigation: perhaps resulting from a learner using a poorly, arbitrarily structured set of nodes or perhaps because a novice is attempting to make sense of material that has been highly structured by an expert.
- Difficulties in integrating information into personal knowledge structures: perhaps, again, this may be the result of a novice having to interact with an expert's knowledge structure or difficulties may arise because the learner is not using the material for any purpose, is not actively engaged with the material and so is only passively, aimlessly browsing.

In addition, a learner's interactions with hypermedia are not predictable, while most instructional design and development models sensibly stress the predictability of learner outcomes from instructional interactions in an attempt to ensure and to perhaps quantify that some learning has actually taken place.

To try to overcome the above difficulties to enable hypermedia to be used most effectively by learners it would seem wise to try to:

- Structure the material in the most natural and appropriate way for the intended learners and to use navigation aids such as maps and metaphors (Preece, 1993).
- Actively involve the user perhaps by requiring them to use the hypermedia for a particular purpose. Also try to gain some control over

the interaction to ensure and to possibly quantify the learning taking place by perhaps using the hypermedia as part of a larger overarching programme or model.

3.1.5. Simulations

Examples of recently available commercial teaching programs built around simulations:

- **Fun School in Time** (PC)
Europress, Tel: 01625 859333, 1995.
Fun School in Time is aimed at the 8 to 11 age band. "Farming", the first section of the program, presents a complex problem-solving scenario in an ancient community. Crops, water and sunshine have to be employed as resources in considered ratios to enable the community to grow. Trade with neighbouring communities needs to be established to achieve development, as well as coping with natural disasters. Another section of the program, "Science", is a complex and engaging application requiring the correct combination of energies to enable a roller coaster to complete its path. The concepts of friction, force, acceleration and gravity all have to be mastered to achieve success.
- **Warwick Spreadsheet** (Macintosh, requires Excel versions 2,3 or 4)
1994
Provides teachers and students with an easy to use yet powerful tool for: simulations, mathematical modelling, interactive diagrams, graphic exploration of numerical data and more. Commands allow modelling of data to be carried out within the system. Teachers can also use pre-written Chemistry, Physics and Biology packs. Teachers can very easily develop their own models and simulations to meet their curriculum requirements and students too can experiment with existing or new data and development projects based on the system.

A computer simulation offers a model of processes or relationships, with which the learner may interact to discover, in possibly altered scales of time and space, without the fear of doing any real harm, what the outcomes are when characteristics are changed.

At the heart of every simulation is a mathematical model and so, only processes or relationships that can be precisely described by equations (and in the process, perhaps, simplified) can be simulated. In addition, the learner can only alter the value of variables in the simulation that are included in the underlying mathematical model. For instance, the relationship between voltage and current in an electric circuit might be simulated. The learner could alter the value of the voltage across the circuit and the simulation program would feed this into the underlying equation, produce a new value for the current and perhaps illustrate

this change by showing a lamp, that was in the circuit, glowing brighter on screen. In real-life, the learner may wish to take the circuit apart and start again with a differently configured circuit. However, in the computer program simulating the circuit this option might not be available.

Systems investigated in the physical sciences such as atoms, planets, machines and environmental processes can all be simulated on a computer because there are equations that describe their behaviour. Similarly, computer based mathematical simulations have also been developed for processes of concern to social scientists such as the economy, population growth and migration.

A variation on the usual computer based simulation is the role play simulation. Role play simulations allow learners to control just one part of a larger model of a human system. For example, in a geography simulation about life in the Sahel (Watson, 1988) each learner takes on a different role. One learner plays the government officer, some play villagers while others still are nomads. Each learner makes their own decisions, for instance, about how much grain to produce or where to water their cattle. These types of simulation are often used when teaching humanities subjects, where the learner has to understand the complex interrelationships between human groups and their environment. Unlike the more typical engineering or mathematical simulations the behaviour of the system is not entirely controlled by the program. Instead, learners discover how other groups of learners behave, how unpredictable the environment can be and how they, as individuals, respond to adversity.

In role play simulations the learner's experience is restricted to the specific role they play so they may have difficulty determining how all the elements in the model interact. This may be overcome by the learner replaying the simulation in another role but this may still result in only partial understanding. Often a full "debriefing" exercise needs to be carried out which brings together the

experiences of all the participating learners.

Simulations allow for discovery learning as a learner can set their own goal or be given a goal by the program and are then able to manipulate the program at will. However, learners are restricted by the specific parameters and options which the software makes available to them. Intrinsic feedback is automatically provided by the simulation as it responds to the learner's actions.

Problems can arise when learners flounder and don't know what to do next. To remedy this, "guided discovery" learning has been developed where assistance in selecting information and defining goals is provided. The result is that often when a simulation is used, there are accompanying "lab notes" providing guidance on what to do and how to do it.

"Learning from simulations is mostly a trial and error discovery process, involving self-generated learning guidance... unless additional external events of instruction are provided by either the instructor or the program."

(Wagner and Gagné, 1988, p46)

However, giving external guidance can lessen the effectiveness of a simulation because learners' are most able to integrate what they are learning with what they already know when they set the goals and guide themselves.

Simulations can be useful but they have to be carefully designed and integrated with other additional learning activities to be most effective.

3.1.6. Microworlds

Examples of recently available commercial teaching programs that can each be described as offering a “microworld”:

- **Fun Physics** (Macintosh)

Knowledge Revolution, 1994.

An entry level physics simulation laboratory where users create experiments by drawing objects on the screen and these experiments are then brought to life with animation. Just about any physical characteristic of an object can be controlled, gravity can be turned on or off and friction and elasticity can be altered. No programming is required.

- **Interactive Physics II** (Macintosh)

Knowledge, 1994.

Combines a simple user interface with a powerful engine that simulates the fundamentals of Newtonian Mechanics. It is a complete motion lab on the computer. Experiments are created by drawing objects on the screen, just like using a painting or drawing program. No programming is required. Springs, ropes, dampers, meters, and a variety of mass shapes are available. Clicking RUN animates the experiment. The powerful simulation engine determines how objects should move while presenting a realistic movie of the experiment.

Papert (1980) is believed to have developed the concept of a “microworld” with his invention of Logo turtle geometry. Microworlds and simulations both provide interactive models for the learner to experiment with and receive intrinsic feedback. However, microworlds strive to deliver an environment where learners can naturally and intuitively learn theoretical concepts, such as Euclidean geometry, Newtonian physics or musical harmony in the same way that they learnt, as children, about the behaviour of “real” world objects such as how objects can be balanced on walls and bounced off floors. Microworlds aim to give the learner direct physical access to abstract theoretical ideas which would normally first require the mastery of a formalism such as Euclidean axioms, mathematics or music theory. Not mastering the formalism can quickly create a barrier to further learning. The microworld is intended to generate

“a new “learning path”... that gets around the block: a computer-based interactive learning environment where the prerequisites are built into the system and where learners can become the active, constructing architects of their own learning”

(Papert, 1980, p122)

The belief is that allowing direct access to the concepts instead of access through a formalism ought to enable learners to learn as freely and as naturally as they do in the real world.

Although academic understanding requires knowledge of the underlying formal theoretical relationships and symbol systems that are used the suggestion is that this understanding is more likely to develop after the learner has informally experimented and is motivated to know more.

White and Horowitz (1991) describe a set of programs they term a “microworld” which they produced to help 12 year olds understand Newton’s laws of motion and develop a scientific way of looking at and reasoning about the world around them. Newton’s laws are particularly difficult for children to understand as they seem to be counter intuitive. For instance, the first law states that “objects do not change their velocity unless a force is applied to them.” From the child’s everyday experience this initially appears to be untrue, for example, if a toy car is given a push and then allowed to freely roll along a floor it will quickly decrease in speed and eventually stop. Of course, there is an unseen force called “friction”, which the child may not be aware of, that is acting on the toy car.

A simulation might teach this topic by reproducing, in a more predictable manner, all aspects of a laboratory experiment. Whereas a more ambitious microworld version might enable learners to go beyond that single experiment by providing learners with the tools to design, build and run their own experiments. The microworld may go so far as to allow learners to not only explore in ways that they can in the real world, but also in ways they can’t. For example, if an equation includes gravity a learner might be able to put in their own value to discover how the experiment would behave if it was being conducted on the Moon or on Jupiter.

Just such an unconstrained, “building block” microworld was developed at Xerox PARC by Randall Smith. The “Alternative Reality Kit” (ARK) enabled the teacher or learner to construct the type of experiment they would normally set up in the classroom.

Generally a computer simulation is built around a fixed model and the learner is therefore only able to learn about this prescribed model. Whereas a microworld would typically make the tools to construct a simulation available to the learners. Using a microworld to construct and carry out experiments can eliminate a number of the real world practicalities that may hinder theoretical understanding. However, at some point, the learners may actually be required to understand and to be able to overcome these practicalities when they are required to conduct their own experiments in the real world.

3.1.7. Comments on styles of CBL

The preceding discussion on styles of computer based learning has outlined the rise and fall of a number of important approaches in the field over the last few decades.

The typical pattern of development within each style has been one of enthusiasm and optimism gradually giving way to the realisation of limitations and then some disenchantment. However, clearly, many benefits can still be seen to remain from each approach.

One of the purposes of this chapter has been to recognise the enduring benefits and pitfalls of the major approaches in order to provide a sound basis for the design of a newer generation of MLEs.

3.2. Initial ideas for the development of the MLE

Initial outline designs included developing a Microworld on the computer that would provide learners with an environment where they could build and test the effectiveness of their own models for dams, spillways and terminal structures. However, this presented a number of problems: the system would take too long to develop due to the complexity of the mathematics involved and the difficulty of providing useful visualisations of the constructed models in operation; the system would, at best, only ever be able to provide an inferior, limited duplication of some of the work that can be done in an hydraulics lab; and many important elements of real world civil engineering project planning, designing and communication might be ignored by the system.

It was also believed that similar drawbacks would arise if a single, large-scale and detailed simulation of, say, a lab experiment or the operation of an existing dam were to be developed.

A second idea was that an effective MLE might be constructed by simply integrating ideas from several styles of computer based learning. In particular, it was proposed to combine the idea of a hypermedia reference base, including animated simulations, multi-choice tests, videos, pictures and so on, with expert systems techniques from intelligent tutoring systems to provide a kind of "expertext" (Rada, 1991) learning environment. Several researchers have combined hypermedia and expert systems techniques including, Brown (1991) "Locator" and Soper and Bench-Capon (1992). Crowder et al (1995) linked a knowledge-based system (KBS) with hypermedia to enable the hypermedia to provide background material on the information given by the KBS. A fault diagnosis KBS was developed (implementation details given in Heath et al, 1994) which was able to evaluate user input and return a list of possible faults. These faults could then be used as a gateway into a supporting Microcosm

hypermedia base. The user was able to follow links from each fault to related information, for instance, a link from a fault might lead to information giving the location of sensors which needed to be tested to determine if that particular fault was the real cause of the problem. The user could also be directed to other question and answer sessions or to other documents in the Microcosm system.

For this research it was thought that it might be useful to link some kind of "dam design" expert system to a hypermedia information base. However, the Journeyman expert was against the idea of incorporating an expert system component which novice learners could use unthinkingly to automatically solve some of the dam design problems they had been given. It was then decided that it might be better to try to use an expert system to filter a hypermedia collection to produce a "map" through the hypermedia material which would indicate only relevant information for a set enquiry within a given context.

This idea of an expert "brain-dump" augmented by multimedia and expert system technology, was particularly attractive to the Journeyman who enjoyed the process of expressing an immense amount of domain knowledge and visualising the vast array of interconnections that existed.

In accordance with this idea a crude hypermedia prototype was developed. A general hierarchic node structure was used, a standard set of link and node types were defined and the format of the contents for each node was determined. Future steps to incorporate intelligent systems techniques may perhaps have involved establishing frames (tables which describe entities as lists of entity attributes and their current values) and ascertaining rules. Frames could have been developed to describe each node and also to describe other elements of interest to the system such as users, contexts and enquiries. Rules could then have been elicited that would have been able to associate the type of user, their current problem solving context and their last enquiry with various nodes so as to

generate useful links and provide a map of relevant information to examine.

However, it soon became apparent, if the MLE were to cover in some depth a significant amount of information, that there would simply be too many cross-references to make the system of any use to a novice no matter what kinds of "intelligent" knowledge screening techniques were used. We were experiencing the combinatorial explosion described by Jackson (1990) in the field of expert systems:

"..it was felt that a memory organisation in terms of constellations of nodes, with spreading activation as the main retrieval process, resulted in a system whose behaviour was insufficiently constrained."

(Jackson, 1990, p160)

It appeared that if a small scale, detailed MLE were to be implemented along the lines described above then only a domain expert would be capable of using it without becoming hopelessly lost and overwhelmed with information. Thus, such a system might appeal to the vanity of experts by allowing them to express the full extent of their knowledge, but it would be of almost no use as a learning tool for the novice.

Although appropriate intelligent filtering techniques could perhaps be developed it was thought that such an approach would be better suited to the development of larger, "coarser" systems which would be aimed at users with more subject expertise. For the development of this MLE it was felt that some means would have to be employed to determine the knowledge to be learnt and some pedagogic strategy would have to be applied to help structure and sequence this knowledge.

In short, it became evident that it was naive to use an approach based on simply attempting to expediently combine elements from different styles of computer

based learning. It was realised that an overall method was needed which initially included a detailed analysis of all the appropriate issues prior to determining the overall teaching strategies and styles of computer based learning to be used. The following section reviews some existing design methods for CBL.

3.3. A brief review of some existing methods for CBL design

Some CBL development methods appear to be more suited to a particular style of CBL. For example, the method described by Ginige et al (1995) is especially suited to hypermedia production while the methods put forward by Bork (1984) and Burk (1982) are particularly applicable to the development of tutorials, but would be less suited to the development of say, an educational simulation. Similarly, each style of CBL can be seen to have a natural set of descriptive techniques associated with it. For instance, the designs for "Tutorial" and "Drill and Practice" programs generally involve flowcharts and structure diagrams; "Intelligent Tutoring Systems" are typically described by frames and rules; "Hypertext/hypermedia" is usually structured according to semantic nets; and "Simulations" and "Microworlds" use mathematical models.

Often, as with the initial approach taken by this developer, it appears that one of the first decisions made by the producer of a CBL package is to determine the style of CBL that is to be used. From the beginning of the development this tends to restrict the descriptive techniques used, the types of knowledge looked for and the pedagogic strategies considered. This author soon realised that it would be better to use an approach which initially analysed in detail what was required before trying to determine the pedagogic strategy to apply and then the styles of CBL to employ.

There follows a brief review of some of the existing methods that are typically used for CBL development. The review is used to indicate the similarities and potential limitations of the existing methods.

Koper (1995) suggests that some developers consider CBL production to be a special case of software development which can be managed with the usual software engineering methods while others consider it to be a special case of

instructional design.

The term 'software engineering' was first used in the late 1960s at a conference held to discuss what was then termed the 'software crisis.' The software crisis arose from the introduction of a new generation of more powerful computers which made it possible to develop far larger and more complex applications than had previously been produced. However, although hardware costs were tumbling software costs were spiralling as existing development methods proved only to be applicable to the production of small systems.

In an attempt to overcome this 'software crisis' and from a belief that software development is essentially an engineering discipline a general model for the software development process was derived from other engineering activities (Royce, 1970). This model was widely accepted as it offered a means of making the development process more visible. There are many variations on this process model which is often referred to as the 'software life cycle.' All of these can be encompassed in the 'waterfall' (because it cascades from one phase to another) model (Sommerville, 1992) whose stages are:

- Requirements analysis and definition
- System and software design
- Implementation and unit testing
- Integration and unit testing
- Integration and system testing
- Operation and maintenance

Sommerville (1992, p. 13) proposes a way of dividing the waterfall process model into more detailed activities, including:

- Requirements analysis
- Requirements definition
- System specification
- Architectural design
- Interface design
- Detailed design
- Coding
- Unit testing
- Module testing
- Integration testing
- System testing
- Acceptance testing

Instructional design (ID) appeared as a separate research area more than 35 years ago. It emerged as psychologists and educators tried to find effective methods for planning and producing instructional systems (Merrill, Kowallis, & Wilson, 1981; Reiser, 1987). Wilson and Cole (1991) suggest that instructional designers have since become recognisably distinct from instructional psychologists working within the paradigm of cognitive psychology (Glaser, 1982; Glaser & Bassok, 1989; Resnick, 1981). For instance, ID theorists tend to stress the development of explicit prescriptions and models for designing instruction, while instructional psychologists tend to concentrate on understanding the learning processes in instructional settings (Wilson and Cole, 1991).

Wilson and Cole (1991) describe how ID models can be divided into two general categories: procedural models for systems design (e.g., Andrews & Goodson,

1980) and conceptual models which incorporate specific instructional strategies for teaching defined content (Reigeluth, 1983, 1987). Instructional strategy models are discussed in more detail in 'Chapter 6. Pedagogic Models'.

The procedural ID models are often depicted by flowcharts representing a series of project phases, progressing from needs and problem analyses to product implementation and maintenance. Procedural ID models tend to be related less to learning theory and more to systems theory and project management methodologies (Branson & Grow, 1987).

Braden (1996) comments on the proliferation of ID models and the ever growing list of alternatives to ID. As long ago as 1972, Twelker et al. reviewed five ID models. In 1980, Andrews and Goodson analysed 40 models. Gustafson (1981), in the first of two surveys on instructional development models, produced a four category taxonomy where each category had a different area of concern, the areas of concern being the: (1) *classroom*, (2) *product*, (3) *system*, and (4) *organisation*. In his second survey in 1991, Gustafson offers a number of examples of ID models which he places in his third category of *system development models*. For example, Dick and Carey's (1990) model which is currently being taught more widely than any other model. The Dick and Carey model consists of the following steps: identify instructional goals, conduct instructional analysis, identify entry behaviours and characteristics, write performance objectives, develop criterion-referenced test items, develop instructional strategy, develop and select instructional materials, design and conduct formative evaluation, revise instruction, and design and conduct summative evaluation. Another *systems development model* noted by Gustafson (1991) is the model by Seels and Glasgow (1990) which clusters steps into the five typical software engineering stages of analysis, design, development, implementation and evaluation (Yang et al, 1995).

Spector et al (1992, p. 47) put forward what they term to be a typical instructional systems development (ISD) model which consists of the following stages and steps:

Analysis	Define training requirements. Analyse target population. Establish performance levels.
Design	Specify instructional objectives. Group and sequence objectives. Design instructional treatments. Specify evaluation system.
Implementation	Implement learning activities. Administer test items. Assess student results.
Maintenance	Revise content materials. Revise test items. Assess course effectiveness.

Barrese et al (1992) have produced CAMCE (Computer-Aided Multimedia Coursware Engineering), their own tailored ISD method developed specifically for the production of multimedia courseware. CAMCE is a systematic approach that is intended to be able to be adapted to different instructional strategies (for example, tutorials or simulations, and so on). The PROFIL (PROduction strategy For Interactive Learning systems) method (Koper, 1995) was developed to integrate instructional design methods with software engineering techniques. It was also intended to be able to take account of the design of courses of which courseware was only a part. Using the PROFIL method a program is designed

and produced in six stages (preliminary investigation, definition, script, technical realisation, implementation and exploitation).

The ISD models discussed above and ISD models in general, as noted by Braden (1996), all make use of and stress the importance of producing performance objectives. However, Spector et al (1992) believe typical ISD models, such as those outlined above, often fail to take account of the relevant cognitive aspects of the material being learnt. They suggest, for example, how a typical ISD model for performing task analysis on a trouble shooting activity might include a description of the particular procedures carried out by the trouble shooter, but it would leave out any account of the mental model that guides the trouble shooter.

3.4. A method for the design of MLEs: the objectives of the research

The research to this point, described in chapters 2 and 3, led to a sharpening of focus of the initial hypothesis. This was that interactive multimedia "has the potential to create learning environments which are capable of elucidating complex conceptual knowledge in ways which have not previously been possible" (p. 6).

The early stages of the work revealed that although the hypothesis seems intuitively to be true, insufficient systematic guidance is available to enable the MLE designer to bring the potential to fruition. There are a number of elements in the design process and its underlying conceptual basis which need to be clarified before interactive multimedia can be fully understood and exploited as an aid to learning. These elements are as follows:

- I. MLEs are capable of showing multiple views of a domain, from many different perspectives. Techniques are needed which identify and describe such perspectives and enable them to be reconciled and combined.
- II. There is a need to make a series of mappings in the transfer of knowledge from sources in the domain to the learner. These mappings are:
 - (a) from the domain source to some form of computer representation, via the interpretation and analysis of the MLE developer
 - (b) from the computer to the learner through a range of media forms, interpreted and understood by the learner.

These mappings require that the MLE designer understands both the problems of the representation of a variety of different types of human knowledge in a computer and the human mechanisms which facilitate the learning of such knowledge.

- III. In order to enable the most effective human learning from computer-held knowledge the designer must know the best form of representation and display to use for each type of knowledge. This requires a clarification of the affordances of interactive multimedia in terms of human psychology.

- IV Learning of individual elements of knowledge is not enough. Learning needs to be integrated into some coherent pedagogic strategy that is capable of making the whole learning experience greater than the sum of its parts. Furthermore, the pedagogic strategy must then be expressed in terms of a learning system architecture which identifies both the software and humanware elements of the learning experience.

Many of these issues can be seen to be similar to issues in the design of knowledge based systems, so that to some extent the design method will resemble knowledge engineering and benefit from some of the techniques and terminology of that medium.

The following chapters, 4 to 8, deal with each of these concerns in turn.

4. Knowledge acquisition

This chapter describes the way in which domain knowledge was acquired for the construction of the MLE. This chapter, and chapter 5, draw heavily on the science of knowledge engineering as the source of the most sophisticated techniques available for the elicitation of knowledge from experts and its analysis in terms of models of knowledge representation. The chapter concludes with a brief review of relevant models of the engineering design process in order to help in the structuring of the learning experience.

4.1. Knowledge sources

The developers, who are not Civil Engineers, were able to compare objectively the relative merits of three distinct views of the domain knowledge and learning needs which were proposed by different categories of expert. While there was little disagreement among the experts as to the core factual content of the domain, there were considerable differences of emphasis which needed to be resolved before a coherent architecture could be arrived at. The different views of the knowledge expressed by our three sources could be mapped separately on to quite different representational paradigms. These are described in chapter 5: "Analysis of the information gained from knowledge elicitation".

Our rationale for drawing upon different levels of expert was based on the work of Kuipers and Kassirer (1984). They report a point related to the general problem identified by Bainbridge (1979) that the expert may simply not verbalise what is "obvious" to him:

"The expert physician, with many years of experience, has so "compiled" his knowledge that a long chain of inference is likely to be

reduced to a single association. This feature can make it difficult for an expert to verbalise information that he actually uses in solving problems. Faced with a difficult problem, the apprentice fails to solve it at all, the journeyman solves it after long effort, and the master sees the answer immediately. Clearly, although the master has the knowledge we want to study, the journeyman will be much easier to study by our methods."

Thus we used in Kuipers and Kassirer's terms, a "Journeyman" as our main expert for knowledge elicitation and a "Master" to validate the knowledge gained. Another reason for this approach was that knowledge elicitation is extremely time consuming, which limits the use of experts of master level.

In total, three sources of expert knowledge were required in order to describe the domain as fully as necessary. The three sources are introduced below:

(a) *The Journeyman*

The expert for this part of the process was a post doctoral Research Fellow with several years of recent experience of hydropower design in industry. He was categorised as a "Journeyman" rather than a "Master". The first part of the knowledge acquisition process involved the journeyman explaining an example design problem involving the design of an energy dissipation device for a dam spillway that was taken from the "Design of Small Dams" (US Bureau of Reclamation (1987), pages 400 - 402). This problem represented a basic illustration of the solution of a typical small scale design problem likely to be given to a novice hydraulic engineer working in an engineering practice.

(b) *The Master*

Secondly, we referred to a "Master", a Professor of Civil Engineering, for confirmation and higher levels of abstraction of knowledge.

(c) *The Industry*

Thirdly, we were influenced by our discussions with industry and ICE Training Managers, who identified a number of practical learning needs relating to job performance in the context of the real world.

For practical reasons it was not possible to consult more than one journeyman or master expert.

Another source of knowledge used were findings from more general research into the process of engineering design.

The remainder of this chapter examines for each of the above knowledge sources in turn how the knowledge was acquired and offers a brief description and rudimentary characterisation of the knowledge that was obtained.

4.2. The "Journeyman"

4.2.1. Knowledge elicitation methods used

A number of techniques from the field of expert system development were exploited in order to elicit from the "Journeyman" the knowledge he used to solve the example design problem. The principal methods employed included a restricted form of Protocol Analysis and the Teachback Interview.

The classical technique of Protocol Analysis involves asking the expert to "think aloud" while solving a problem; this is tape recorded, transcribed in detail (along with notes of any actions) into a "protocol", and then analysed for meaningful associations. The interviewer intervenes during the task performance only with non-directive reminders to keep thinking aloud. The effects of verbalisation upon task performance have been examined by Ericsson and Simon (1984), who found that, provided the subject was not asked to explain their thoughts, instructions to think aloud do not significantly alter the sequence of cognitive processes.

This whole process of protocol analysis is very time consuming (Neale, 1988). A shortened version was used consisting of the expert solving a problem step-by-step while writing down the solution and talking aloud with the interviewer taking additional notes.

The principal merit of protocol analysis is that in deriving from a much more true-to-life task situation than any interview method, the unconstrained verbalisations

of the expert may reveal the knowledge (particularly the heuristics) that experts use in problem solving, which they are unable to articulate in an interview. This knowledge usually has to be inferred from the data. Welbank (1983) suggests that protocol analysis is most often used to verify knowledge derived from theory, which makes it a useful check against the validity of textbook procedures, or for comparison with the reasoning methods which experts say they use, which makes it a useful technique to use in conjunction with the Teachback Interview.

Teachback Interviewing is based on the Conversation Theory of Pask (1975). The technique is distinguished from other interviewing methods by being a participant activity rather than one between an interrogator and a respondent (although the principle is identifiable in the stress placed by Spradley (1979) on "restating what the informant says, in his own words". It is very much "expert driven", in the sense that any topic to be discussed must originate from the expert, who has to specify at least one link with another topic.

The expert describes a procedure to the interviewer, who then "teaches" it back in the expert's terms and to the expert's satisfaction. When they agree, it is Pask's Level 0. At Level 1 of the interaction, the interviewer asks the expert how the concept was reconstructed, and the Teachback procedure continues until the expert is satisfied with the interviewer's version. Then the interviewer has understood the expert. Level 1 is concerned with explanations of explanations or knowledge about knowledge, i.e. metaknowledge. Strengths of the technique include the relative absence of any preconceptions about the domain which the interviewer might have, its non-judgemental nature, and its success in gaining

and retaining the expert's interest. On the other hand, it is not a strongly structured technique, is tiring for both participants and can produce a mass of transcribed information. This has encouraged the use of short interviews (usually approximately one hour in length).

The achievement of understanding (by whatever means) and its expression in a publicly examinable form is central to the knowledge acquisition process (Neale, 1988). Words are less explicit and less reliable as a representation of the mental model of either participant than a graph, diagram or other pictorial representation (Regoczei and Plantinga, 1987). Both the content and the meaning of the knowledge conveyed by the expert must be shared by the interviewer. To document the knowledge elicited flowcharts, decision trees, semantic nets and hierarchic forms of entity relationships including inheritance features (frames) have all been used, typically, iteratively to validate and expand understanding. Another important technique which has been used in the same way and to retain the expert's interest is "rapid prototyping". This was initially carried out on paper and then later developed using the appropriate software tools. However, whilst frequently referring the emerging system to the Journeyman for comments has its value it was recognised that the Journeyman's time is precious and should be treated with respect and not wasted by, for instance, embroiling him with technical matters such as the intricacies of the authoring language or minor evolving design details.

At the beginning of the knowledge acquisition process, during what has been called "orientation" (Breuker and Wielinga, 1987), a limited kind of "Domain

Definition Handbook" (Grover, 1983) was developed which contained:

(i) *Bibliography of Principal References.*

Although formal textbook models of expertise can vary considerably from the way experts practise, it was realised that reading relevant textbooks is often the quickest route to a general understanding of the domain (e.g. Roycroft and Loucopoulos, 1985). In fact, it has been suggested that in the process of learning to become an expert the problem-solving and knowledge-structuring skills essential to an expert are often developed from the idealised knowledge of textbooks (Bobrow, Mittal and Stefik, 1986).

(ii) *Glossary of terminology.*

Many authors observe that a fundamental prerequisite for effective knowledge acquisition is a degree of mastery over the vocabulary of the domain (e.g. Breuker and Wielinga, 1984; Kahn, Newlan and McDermott, 1985). To be unaware of basic concepts and terminology is to risk inhibiting the expert, or even alienating him or her by requiring him/her to spend valuable time on explanations (Welbank, 1983). In interacting with the expert, the same terminology as the expert should be used to avoid errors introduced by the expert translating questions into more familiar terms (Bainbridge, 1979); this may not always coincide with textbook terminology. So, to avoid confusion, an alphabetic index of expert terminology was produced and was continually expanded and updated throughout the knowledge acquisition process.

4.2.2. The example design problem

The problem used as a vehicle for knowledge acquisition was believed to represent a typical small scale design problem likely to be given to a novice hydraulic engineer working in an engineering practice. It was taken from the "Design of Small Dams" (US Bureau of Reclamation (1987), pages 400 - 402). The problem embodied the design of an energy dissipation device for a dam spillway.

By its very nature, the streamflow which supplies a dam reservoir is variable. It follows that there will be times when the reservoir is full and the streamflow exceeds the demand. The excess water must therefore be discharged safely from the reservoir. In many cases, to allow the water to simply overtop the dam would result in a catastrophic failure of the structure, with the potential for loss of life. For this reason, as illustrated in Chapter 2, 'Section 2.1. The subject matter', carefully designed overflow passages - known as 'spillways' - are incorporated as part of the dam designs. The flow discharged from a spillway often attains high velocities. If this flow were left uncontrolled, severe erosion at the toe of the dam could occur. Therefore, it is necessary to dissipate much of the energy, and return the water to the depth and velocity appropriate to the stream below the dam. This is usually achieved by placing a dissipating or "stilling" device as a terminal structure at the base of the spillway. Typical terminal structures are:

- (a) stilling basin;
- (b) plunge basin;
- (c) submerged bucket; and
- (d) ski jump/deflector bucket

Details of the actual example design problem and some of the material elicited from the Journeyman during its solution are given in Appendix I: Solution to the

elicited example design problem.

The Journeyman solved the example design problem by selecting and refining standard, well-understood "off-the-shelf" civil engineering structures to fit the given circumstances. Initially, values relating to the existing dam and spillway structures (e.g. crest elevation and crest length) as well as relevant environmental data values (e.g. streamflow into the reservoir with corresponding maximum and minimum spillway discharge; and tailwater and ground conditions) were all analysed. These values were matched to the characteristics offered by different types of energy dissipation device to select suitable device types. Stilling basins and submerged bucket dissipators were deemed most appropriate. Then, further analysis, led to more specific versions of these devices being chosen. i.e. a US Bureau of Reclamation Type III stilling basin was singled out and a slotted bucket design was chosen as opposed to a solid bucket design for the submerged bucket dissipator.

Various characteristics of the selected specific device were then balanced with each other to satisfy the competing constraints. The constraints, i.e. some example values already mentioned, also included ensuring the spillway water's energy is dissipated and its level matches the varying depths of water in the stream below the dam for the whole range of likely flow, including floodflow.

Problems such as the example design problem can be characterised as problems of "Routine design" (see Dym and Levitt, 1991). That is to say, in routine design there are effective problem decompositions, compiled plans for designing the components, and failure analysis information that can be usefully applied. These problems require significant amounts of design domain knowledge because of complex interactions between subgoals and between components, as a consequence of which the complexities both of plan selection and ordering and of backtracking to undo failures must be anticipated. Thus,

even in this "simple" class of routine design, there is more than ample scope for deploying knowledge (Dym and Levitt, 1991).

Trainees will have to be able to solve such routine design problems before they can move onto "Creative design". Creative design is characterised by goals that are vaguely specified, a paucity of effective problem decompositions, and a scarcity of designs for sub problems. This kind of design requires considerable problem solving even in its auxiliary processes. This type of design is innovative, rare, and extremely difficult to teach - largely because we do not understand the origins or form of true creativity (Dym and Levitt, 1991).

4.2.3. Description and characterisation of the knowledge obtained

The example design problem is typical of the domain in terms of the breadth and depth of information it entails. Analysis of the information identifies four knowledge categories: (a) *physical structures*; (b) *hydraulics*, i.e. the mechanical behaviour of water in the structures, expressed as mathematical models; (c) *environmental data*, which describes the geographical, geological and hydrological context of the dam's environment and (d) *procedural knowledge*, i.e. active knowledge of the appropriate steps and skills to be used to solve the problem. The structure of the information within each category will now be discussed:

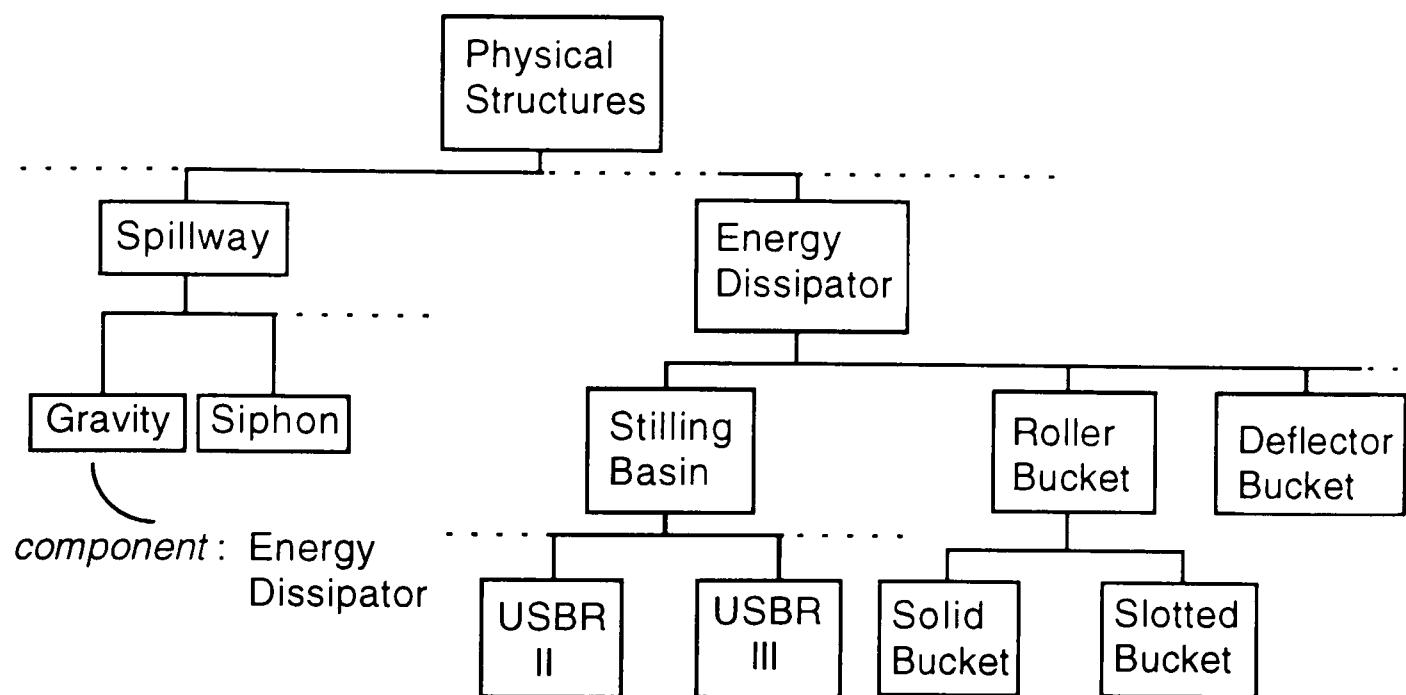
(a) *physical structures*

Structure types can be differentiated by their various characteristics e.g.:

- what they are, i.e. allowable physical characteristics: shape, dimensions, material
- what they do, i.e. their range of function
- when and where they should be used, i.e. range of suitable operating conditions
- how they may be decomposed into definable, discrete yet interacting components which are themselves substructures.

The above attributes can be used to classify widely recognised and distinguishable types of physical structure. Knowledge of a unique type's range of function or suitable operating circumstances may be based on detailed mathematical modelling, empirical testing of real world structures or past expert experience (rules of thumb).

Figure 7. Organisation of physical structures.



USBR II - United States Bureau of Reclamation Type II Stilling Basin^{*1}

USBR III - United States Bureau of Reclamation Type III Stilling Basin^{*1}

*1 - from: US Bureau of Reclamation (1987) pages 388-398.

Figure 7. uses the example of types of spillway and types of energy dissipator to illustrate how distinct types of physical hydraulic structure can be classified.

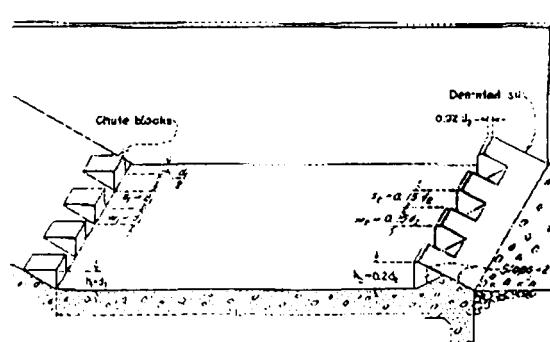
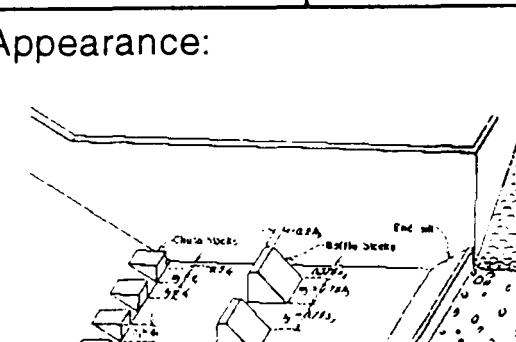
Analysing the branches of the classification tree underneath the "Energy Dissipator" node, reveals that both of the structure types: "Stilling Basin" and "Roller Bucket" share the same set of attributes, although each may refine or modify some of those attributes, which are present in the more general structure type: "Energy Dissipator". That is, "Stilling Basin" and "Roller Bucket" can each be said to be *a kind of* "Energy Dissipator". Similarly, "USBR II" and "USBR III" share the same attribute set (some of which may, again, vary) that are present in "Stilling Basin", that is, they can both be said to be *a kind of* "Stilling Basin". Also, the types: "Solid Bucket" and "Slotted Bucket" can both be thought of as *kinds of* "Roller Bucket". In this way, a physical structure can be examined in varying

degrees of detail, i.e. a preliminary study of the generic type: "Energy Dissipator" may involve considering only the general types: "Stilling Basin" and "Roller Bucket" whereas a more detailed investigation might also examine the specific types: "USBR II" and "USBR III".

As stated, above, a structure may also be said to consist of discrete, separately definable yet interacting components which are themselves structures. For example, as shown in Figure 7, a "Gravity" spillway can be said to need an "Energy Dissipator" (i.e. a "*has a*" relationship; Rada, 1991). In this case, the "Energy Dissipator" might be a type of "Stilling Basin" or "Roller Bucket". In this way, the use of *classification trees* for all levels of physical structure, from types of "Hydropower Project" to types of "Chute Block" (a bottom level component of some types of "Stilling Basin") enables any size hydraulic design project to be represented.

Thus, most knowledge relating to physical structures can be seen to be declarative and can often be organised into inheritance hierarchies as shown in Figure 8.

Figure 8. Organisation of physical structures including attributes¹

Structure Name:	Hydraulic Jump Stilling Basin
Plan:	Rectangular
Froude Number:	_____
Incoming Velocity:	_____
Components:	_____
Appearance:	_____
Types of:	USBR I, USBR II, USBR III, USBR IV
.....
Structure Name:	USBR II
Plan:	Rectangular
Froude Number:	> 4.5
Incoming Velocity:	> 18 m/s
Components:	Chute Blocks, Dentated End Sill
Appearance:	
Structure Name:	USBR III
Plan:	Rectangular
Froude Number:	> 4.5
Incoming Velocity:	< = 18 m/s
Components:	Continuous End Sill, Chute Blocks, Baffle Blocks
Appearance:	

*1 - Values from: A. Lencastre (1987).

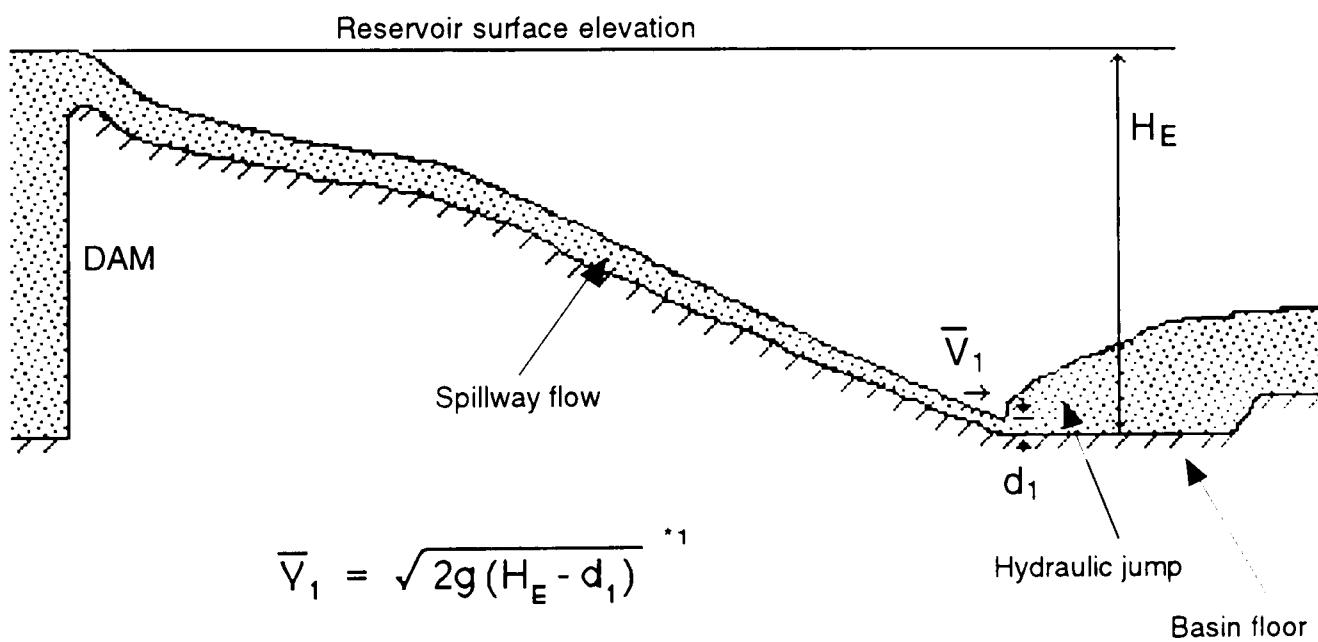
Diagrams from: US Bureau of Reclamation (1987).

(b) *Hydraulics*

The behaviour of water flowing over physical structures is described by

mathematical equations which account of the key parameters. For example, rate of flow down a spillway is a function of gradient, cross sectional area, smoothness of bed, length, and so on. Specific equations or rules exist to describe all conventional situations, and these must be accessible as required by the learner, for example Figure 9.

Figure 9. Example equation: Incoming velocity at hydraulic jump stilling basin floor level.



\bar{V}_1 = incoming velocity at basin floor level.

H_E = specific energy: reservoir water surface elevation minus basin floor elevation, assuming no loss in specific energy.

d_1 = upstream depth of flow at basin floor level.

g = gravitational constant.

*1 - US Bureau of Reclamation (1987), page 402.

(c) Environmental Data

The engineer has to be aware of the physical context of the site, i.e. local hydrology, geology and geography. Typical data i.e. reports and surveys on ranges of existing conditions are accessed to illustrate the variables affecting

design problems, for example figure 10. Environmental data can be seen as lying on the boundary of the knowledge that the problem requires.

Figure 10. Example environmental data: General storm dimensionless S-graph data, Rocky Mountains, describing the actual flows following rainfall.^{*1}

This table describes the behaviour of the site region with respect to the way in which rainfall results in water runoff into streams and rivers.

Time, % of L_g	Discharge % of ultimate
5	0.05
10	0.23
...	...
95	47.3
100	50.00
...	...
195	79.43
200	80.26
...	...
595	99.95
600	100.00

L_g = Lag time, the time from the start of a continuous series of unit rainfall excess increments to the time when the resulting runoff hydrograph reaches 50% of the ultimate discharge.

^{*1} - US Bureau of Reclamation (1987), extract taken from the full table on page 45.

(d) *Procedural Knowledge*

The design of structures requires knowledge of physical structures and hydraulics, but these in themselves are relatively passive forms of knowledge. Design is an active process which also requires knowledge of the steps and skills in the procedures of solving the problems of defining how a particular

structure should be built to meet a particular set of needs. Such procedural knowledge can be thought of as being the set of rules which the designer follows in moving from problem to solution. In specific terms this involves the steps required for making decisions and choices about layout, types of component, materials and so on. Baker et al (1991) provide such a cognitive model of (electronic) engineering design. Figure 11 shows how it could be applied in Civil Engineering.

Figure 11. Simplified pseudo code example of a procedural confirmation.

The code here would be called from within the control strategy code in Figure 12. It uses the Incoming Velocity parameter (defined in Figure 9.) and the Froude Number parameter (mentioned in Figure 8.) to identify the suitability of the USBR Type III Hydraulic Jump Stilling Basin (discussed in Figure 8. and Figure 7.).

```
FUNCTION IsASuitableMoreSpecificStillingBasin stillingBasin
IF the calculated Froude Number is within the range of the value in
    the slot (Froude Number) of the node with stillingBasin in the slot
        (Structure Name)
AND
    the calculated Incoming Velocity is within the range of the value in
        the slot (Incoming Velocity) of the node with stillingBasin in the slot
            (Structure name)
THEN
    RETURN TRUE
ELSE
    RETURN FALSE
END IF
END IsASuitableMoreSpecific
```

Wilson, B., and Cole, P. (1991) also refer to control strategies which are required for problem solving activity. These control strategies have monitoring, diagnostic and remedial components and are used by the problem solver to monitor and regulate their problem solving activity. For instance, during solution of the example design problem, the problem solver has to be aware of and be able to apply the steps necessary to confirm the suitability of a chosen type of stilling basin. A basin's failure to conform must be recognised and the appropriate

remedial action taken, such as selecting and examining a more eligible type of basin. At a higher level, designers exhibit generalised strategies for designing, for example, problem decomposition. Metaknowledge of this type is illustrated in Figure 12 and a more general review of the role of models of engineering design appeared in Chapter 4, 'Section 4.5. Review of relevant models of the engineering design process.'

Figure 12. Simplified pseudo code example of a problem control strategy.

This code calls the function defined in Figure 11.

```
PROCEDURE ExamineMoreSpecificStructures
IF the current node is NOT a leaf node
THEN
    set aMoreSpecificStructureFound to FALSE
    REPEAT with number = 1 to the number of elements in slot (Types of ) of current node
        IF IsASuitableMoreSpecific & (slot (Structure Name) of current node) ((element (number) of slot (Types of) for current node))
        THEN
            Examine & (element (number) of slot (Types of) for current node)
            set aMoreSpecificStructureFound to TRUE
        END IF
    END REPEAT
END IF
IF aMoreSpecificStructureFound = FALSE
THEN
    TakeRemedialActionForAMoreSpecificStructureNotFoundOn-
        Node (slot (Structure Name) of current node))
END IF
END ExamineMoreSpecificStructures
```

Examples of knowledge from the foregoing four categories were all identified as being needed to support the target learner. Supplied by the "Journeyman" it was confirmed by the "Master" and industry experts as being correct and relevant to the needs of the target learner, and constitutes the core knowledge of the system.

4.3. The "Master"

4.3.1. Knowledge elicitation methods used

A number of informal discussions were held with the "Master" where he conveyed his opinions on the approaches taken to the solution of problems in the domain, formed from many years of teaching experience and extensive industrial and consultancy experience.

4.3.2. Description and characterisation of the knowledge obtained

The contribution of the Master expert, a Professor of Civil Engineering, was much smaller in volume than that of the Journeyman, but nevertheless, highly significant. First of all he provided confirmation of the Journeyman's approach to problem solving and the knowledge content of the domain, as expected.

High Level Conceptual Knowledge

Secondly, he emphasised a higher level perspective of the domain knowledge which clarified some important conceptual issues in the area of hydraulics. The Journeyman's account of hydraulics provided all of the equations necessary to describe the behaviour of water flowing over the spillway structure, as described in section 4.2.3. It also showed some of the common origins of those equations, and the way in which one equation may be modified to satisfy different requirements. The Master was able to generalise these equations even further to show how they all derive from one or more of only three fundamental laws of physics, namely:

- conservation of matter (conservation of mass)

- conservation of energy
- conservation of momentum

This perspective provided us with a top level to the hierarchy of hydraulics concepts being considered and thus a fundamental conceptual closure for the learner.

Experiential Knowledge

The Master's third contribution was equally fundamental. It arose in discussions of the behaviour of tailwater, that is the water discharged by the dam spillway into the bed of the original stream. Because of wide variations in the erodability of stream beds in different situations, the mathematical models of their behaviour are extremely complex, costly to create and of variable accuracy. However, design engineers must still be able to predict the performance of the system as a whole, including that of the stream bed. If they do not, then they run the risk of designing a spillway whose terminal structure is inadequate to calm the flood flow to the point where it does not erode the stream bed. In the worst case the stream bed will be washed away, taking the dam with it (see Figure 2, Erosion at the toe of a spillway). Excessively cautious design, on the other hand, could be much more expensive than necessary.

In situations like this the engineer may construct a physical model in the laboratory to try various designs, or consult a higher level of expert who has had previous experience of similar conditions. Although this particular example is probably in the realms of "creative design" rather than "routine design", it does raise the question of how the MLE should cope with ill-defined, experiential knowledge.

4.4. The "Industry" view

4.4.1. Knowledge elicitation methods used

Our third and final view of the learning domain was derived from experienced managers in the Civil Engineering Industry and from the Training Manager of the Institution of Civil Engineers. It was acquired by informal interviews and consists mainly of beliefs and intuitions about what is needed for graduate training in the domain. Although we would have liked to conduct a wider and more formal survey of industry views we found that the prevailing economic climate for civil engineering resulted in a very poor rate of response.

4.4.2. Description and characterisation of the knowledge obtained

There was a high degree of agreement among those we consulted as to the shortcomings fresh graduates in professional training. Their approach to design was characterised in the following phrases:

"they plunge into the deep end and try to get to the equations as soon as possible."

".. no consideration of options and alternatives in the early stages"

"A design should look right and feel right in broadly sketched terms. New graduates have no concept of this feel for an appropriate design."

"They have no drawing skills. They don't sketch out their ideas at all."

"They lack the ability to stand back from a project and consider it as a whole."

"They lack cost awareness - they need to think money."

"They have no grasp of the effect of the forces they are dealing with in real world situations."

"Engineers spend much of their time writing letters and reports in the normal course of design. We are in frequent communication with customers, contract holders and sub-contractors at different stages of the project. Graduates don't know how to write these things to a professional standard or realise how important they are."

These views have been summarised as follows:

- I Novice designers follow a depth-first approach, whereas a breadth-first approach is required for early stages of design.
- II They have little or no model of the physical magnitude and effects of forces in full-scale hydropower projects.
- III They have a poor model of the task structure of design. It does not include a grasp of the phases of a design project, of key communication skills, or of business issues.

Points (I) and (II) above were largely expected at the start of the project, (II) in particular was one of the main motives for using multimedia, as described in Chapter 2. However point (III) was unexpected, and had not been indicated by either the Journeyman or Master experts.

The implication of point (III) for the design of the MLE is that it should teach, or at least reinforce, a model of the design process and the skills associated with applying such a model.

The traditional six year apprenticeship of graduate civil engineers should be

expected to equip them with very clear and rich models of both the design process, and the task structure of projects, i.e. structured, clearly defined methodology of the design process, including reviews, client meeting and communications, as well as appropriate communication and business skills. However, six years is a very long time. An important objective of the MLE is to accelerate this learning. Furthermore, metacognitive knowledge is equally important for undergraduate learners who have less exposure to industrial practice.

4.5. Review of relevant models of the engineering design process

The strategies and heuristics needed to accomplish the design tasks within the project life-cycle were identified as important types of knowledge that would have to be conveyed by the MLE. The "Journeyman" seemed to employ a number of design strategies as he worked through the example design problem. Confirmation for, and more detailed and comprehensive characterisations of, such strategies was looked for in the research literature. The research examined, in particular, included: Jefferies et al's (1981) investigation of the design schemata of novice and expert Software Designers; Ullman et al's (1988) model of the design process followed by Mechanical Engineers; and, Ball's (1990) study of the design practice of semi-expert and expert Electronic Engineers. The goal was to uncover a number of relevant design strategies from accepted models of the design process. These strategies, which exemplified good or expert design practice as opposed to poor or novice design practice, could then be incorporated and conveyed within the MLE, and also help to provide a sound structure for the delivery of the components of the learning material.

Jefferies et al (1981)

A study by Jefferies et al (1981) investigated the design schemata of novice and expert Software Designers. Their findings for both novice and expert designers can be summarised as:

- (1) Experts
 - (a) Divide the problem, typically in a top-down and breadth first manner, into manageable and minimally interacting subproblems.
 - (b) Understand the problem before breaking it into subproblems.
 - (c) Retrieve a known existing solution, or adapt a known similar solution to the particular context of the problem.

- (d) Remember detailed and well-integrated representations of in progress solutions and are able to retrieve them when needed.

(2) Novices

- (a) Try, with difficulty, to decompose the problem into various levels of subproblem. The usual result is a top-down and depth-first decomposition.
- (b) Risk design correctness and efficiency by rushing the solution of a subproblem without first gaining a real understanding of it or exploring its different aspects.
- (c) Are unable to apply recent classroom work to solving the subproblems.
- (d) Cannot integrate information generated during earlier stages of the solution attempt with work done during later stages.

The expert's years of experience enable the procedures of the schema to become automatic, freeing the designer to focus more on the details of the specified problem and allowing more complex problems to be solved.

Ullman et al (1988)

Ullman et al (1988) provide a detailed model of the design process followed by mechanical engineers. The task/episode accumulation model (TEA model) which they developed was based on a two-year study of the mechanical design process and was developed from the detailed analysis of the audio and video protocols of five experienced mechanical design engineers solving real design problems.

At a basic level the design process is characterised as the application of various "design operators"

"design operators are primitive information processes that modify the design state by performing calculations and simulations, creating new proposed designs, evaluating proposed designs and making decisions to accept or reject proposed designs. The TEA [task/episode accumulation] model contains ten operators select, create, simulate, calculate, compare, accept, reject, suspend, patch and refine."

(p34)

The application of a string of a suitable "design operators" constitutes a meaningful design "episode."

"An episode is a sequence of operator applications that addresses some primitive goal. The nature and scope of primitive goals changes as the design unfolds."

(p35)

Within this model, the completion of a mechanical design project is perceived to involve the solution of a hierarchy of goals. The top level goal is to produce a design that satisfies the given constraints. Below this Ullman et al describe three general categories of task level goals:

- (a) Conceptual design in which the major forms of the solution are conceptualised to satisfy the major constraints.
- (b) Layout design in which the components are specified at decreasing levels of abstraction.
- (c) Detail design in which attention focuses increasingly and gradually on the refinement and documentation of the components.

Ullman et al describe this general goal structure as a progression from abstraction, through an intermediate stage to the concrete, each stage having its own type of notation of the form verbal/textual, visual or physical.

Ball (1990)

Ullman et al's model of the design process can be seen to be essentially technical and solitary. Ball (1990), describing a protocol study of the design processes in Electronic Engineering, identifies a social dimension to design, illustrated in Figure 13.

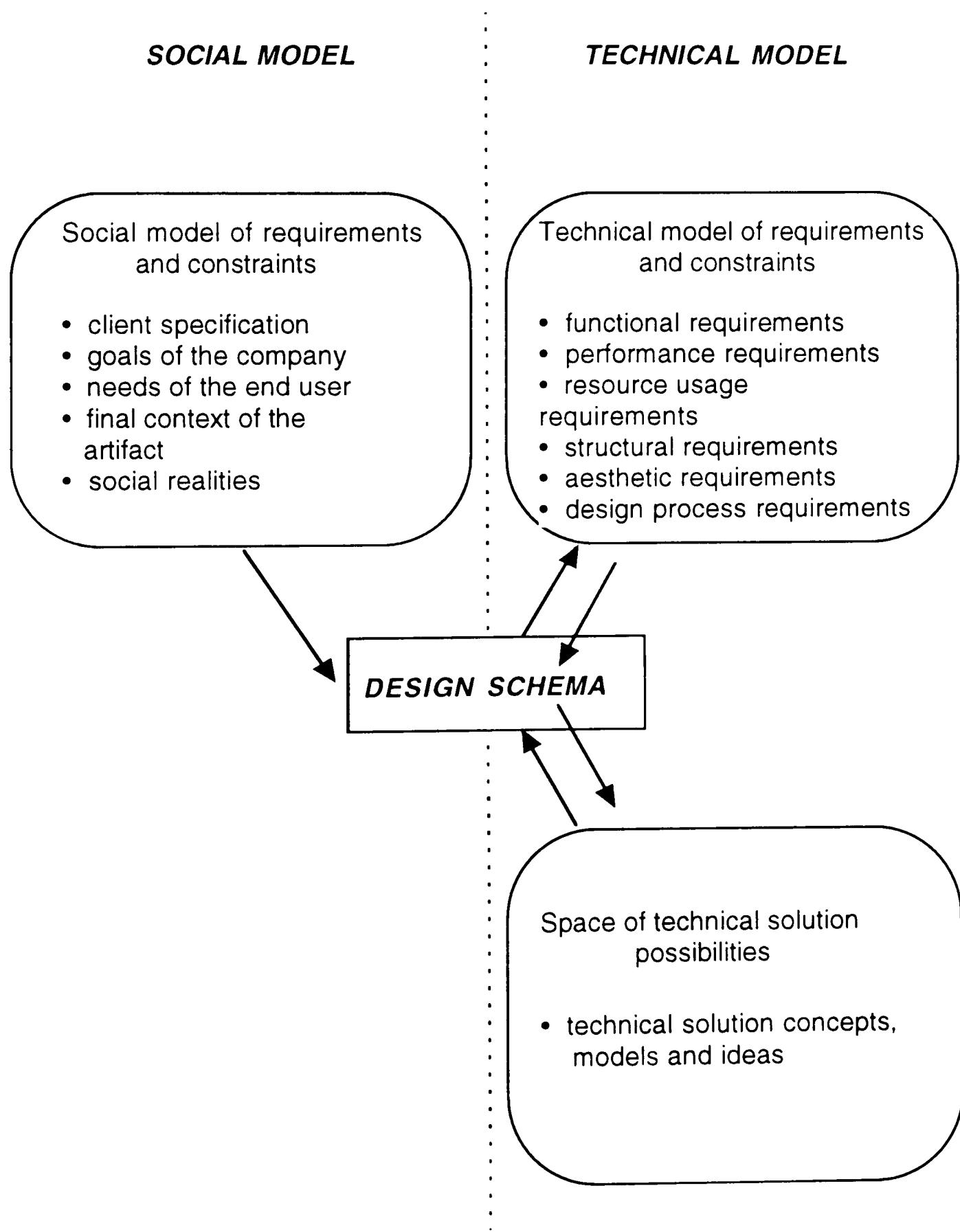
The importance of the social dimension to design is also recognised in Software Engineering where models of the life cycle of the design process usually include specific mechanisms for communication between designers, their managers and their clients, (for example, Somerville 1992, chapter 1) and mechanisms for communication within the project team (for example Somerville 1992, chapter 2).

Ball's (1990) work studied two groups of electronic engineers, semi-expert and expert. Each group was monitored as they attempted an engineering design task.

Observations of **semi-expert** engineers, revealed:

1. Movement between sub-problems.
 - (a) A "problem reduction strategy" was used by the students to produce a "top down" hierarchy of manageable and minimally interacting functional modules.
 - (b) The modules were dealt with sequentially in a depth-first manner down through the hierarchy.

Figure 13. Mapping between the social model and the technical model via the design schema. (Externally imposed requirements and constraints are mapped onto the technical model of the problem by the design schema which in turn controls the search for technical solution concepts). (Ball, 1990).



2. Development and evaluation of design solutions.

- (a) The students adopted a "satisficing" (Simon, 1969) strategy. A single high level solution, which was deemed to be satisfactory, would be chosen and then worked on in depth instead of developing and comparing a number of alternative high level designs with the aim of producing an optimum design.
- (b) Although the students did not apparently consider alternative high-level solutions having found one that sufficed, Ball observed that they did refine their designs at lower levels in the design hierarchy. This, he suggested, indicates that the students did demonstrate some inclination to optimise but only in the lower levels of the hierarchy. Ball does not make clear whether these refinements were made because the original design did not meet the specification and was thus unacceptable as it was, or whether the design actually satisfied the requirements and the refinements were improvements to this workable design. The former would be designing and the latter optimising.
- (c) There was evidence to show that the students were constructing and manipulating mental models to simulate the functioning of their designs. This is important for comparison and evaluation of different designs.

Observations of **expert** engineers, revealed:

1. Movement between sub-problems.

- (a) Again the engineers employed a "top-down" hierarchy of modules.
- (b) Unlike the semi-expert engineers, the hierarchy was traversed in a breadth-first manner.

2. Development and evaluation of design solutions.

- (a) Again the engineers employed a "satisficing" design strategy.
- (b) Again the selected design concepts were refined and improved at the lower levels.

Overall, the design strategies were very similar within this group of expert engineers, which in turn shared similarities to those of the semi-expert group.

Although the studies used electronic engineers as subjects, Ball believes that the findings may be generalised to encompass the design strategy of all designers regardless of their particular domain.

It is interesting to note that Ullman et al. (1988) also report that, "Our designer subjects are satisficers, not optimisers". With this in mind, Ullman et al. go on to emphasise the need for a good conceptual design in the early stages of design because all subsequent effort involves refining and patching this basic idea.

Initially, much of the meta-knowledge of the type described in this section was observed in the Journeyman rather than being made explicit by him. It was articulated mainly as a result of "Teachback" during knowledge elicitation as discussed in section 4.2.1. Some of the work discussed in this section was extremely useful as the level of detail employed enabled the researchers to uncover and label a number of working methods, skills and knowledge types which seemed to correspond to those which it was noticed were being employed by the Journeyman as he solved the example design problem. For example, while working on the example design problem the Journeyman was seen to apply a number of Ullman's TEA model operators, namely: "select," "calculate," "compare," "accept" and "reject." These and other elements of the TEA model are

represented in different parts of the terminal structure design process.

Similarly, as discussed earlier in this chapter, many of the failings in the general strategies, skills and emphasis employed by inexperienced engineers in the studies in this section were also suggested by the Master and those in industry as being some of the weaknesses typically displayed by students and recent graduates.

The findings from the studies reviewed in this section can be seen to support the elicited strategies discussed earlier in this chapter. In conclusion, it can be seen that the framework provided by the MLE should address a number of issues. First it should provide a clear model of design as a top down process starting with a breadth first consideration of alternatives at a conceptual level, selection of one alternative, culminating in detailed design of specific components. Secondly, it should acquaint designers with the appropriate notations for each stage. Thirdly, it should reinforce a social model of design in which clients, managers and co-designers are all recognised. Fourthly it should indicate appropriate forms and notations for communication between these parties.

4.6. Methodological implications arising from knowledge acquisition

Several aspects of the knowledge elicitation process described in this chapter and its resulting perspectives, can be considered in relation to the methodology of the design of computer based learning, particularly interactive multimedia:

- (1) Using knowledge elicitation techniques, such as "teachback" from knowledge engineering can enable domain novices (who may be specialist multimedia developers) to develop a systematic, verified description of the knowledge domain and the practical techniques and heuristics used by domain experts, without the full time assistance of domain experts. This approach may offer advantages over the more commonly described situation in which domain experts develop computer based learning systems with a limited grasp of interactive multimedia technology.
- (2) Using a range of experts with different contributions results in a richer, better structured and more clearly defined view of the domain knowledge than a single expert's view, in this case including:
 - (a) conventional declarative "text book" view (journeyman)
 - (b) problem solving strategies (journeyman)
 - (c) high level conceptual view (master)
 - (d) experiential knowledge (master)
 - (e) project structure and management (industry)

Conventional approaches to learning needs analysis tend to focus mainly on knowledge type (a), conventional declarative knowledge.

- (3) The role of experiential knowledge in particular, and its nature which makes it very difficult to express, say, in text or equations, has particular implications for multimedia. Knowledge which derives from sensory experience, for

example the erosion behaviour of stream beds, may often be able to be captured and expressed in multimedia, enabling it to be taught widely in ways which, for practical reasons, have not previously been possible.

- (4) Knowledge elicitation is not a perfect science, particularly if only one expert of each type can be interviewed. For example, there is a risk of personal bias or influence from an individual cognitive style. However, those risks are present to an even greater degree if some form of knowledge elicitation is *not* used and only one expert source is accessed. The latter situation is often the case in the literature of computer based learning where domain experts have been responsible for the design of such systems.

In this research it was noteworthy that little or no contradiction of one expert by another occurred. Successive experts brought confirmation of what had been heard before and, more importantly, additional insights into the domain.

It is also important to be aware that the techniques of knowledge elicitation used can, to some extent, shape the style in which the expert's knowledge is presented. For example retrospective rationalising about the design process does not provide the same model of events as say, video protocol analysis. It is desirable to confirm the validity of the models offered to the learner with independent evidence. In this case we referred to models of engineering design offered by Ullman et al (1988), Sommerville (1992), Ball (1990) and Baker et al (1991) all of which were based on extensive studies of design protocols.

- (5) Implicit knowledge:

Knowledge elicitation techniques such as teachback and protocol analysis can reveal experts' implicit knowledge. The quality of knowledge elicitation can also be enhanced by the use of published studies of the domain.

particularly where these describe knowledge which would normally be implicit in experts' accounts, for example, the meta-knowledge contained in the literature on the design process (Ullman et al (1988), Sommerville (1992), Ball (1990) and Baker et al (1991)).

The existence of a range of fundamentally different types of knowledge in the learning domain raised questions as to the way in which the MLE should store knowledge and express it to the learner. Chapter 5, following, provides an analysis of the underlying nature of these different types of knowledge and the ways in which they are thought to be understood by learners.

5. Models of human knowledge: analysis of the information gained from knowledge acquisition

Further analysis of the elicited domain knowledge, which was roughly characterised in the previous chapter, revealed a number of different aspects to the knowledge. Each of these aspects seemed to require their own methods of representation in the knowledge base of the MLE, and crucially, their own styles of presentation to the learner.

Analysis uncovered three clearly distinct categories of knowledge, as follows:

- (i) *Formal, symbolic knowledge such as rules, facts and equations.*
- (ii) *Experiential knowledge not describable in formal terms but visualisable via video or animation.*
- (iii) *Metacognitive knowledge concerning the application of formal and experiential knowledge at the appropriate moment.*

This chapter describes then provides evidence from reviewed literature to substantiate, in turn, each of the above knowledge categories. The chapter concludes by summarising the characteristics of each category and outlining the most appropriate presentation styles that the MLE can use to present the knowledge within each category to the learner.

5.1. Symbolic or rule-based

The Journeyman's account of the core knowledge, as confirmed by the Master, is characterised by its analytical, reductionist approach. The elements of the solution to the example problem are decomposed into categories i.e. physical structures, hydraulics environmental data and procedural knowledge. The elements in turn seem to lend themselves to further decomposition, for example, the inheritance hierarchy of physical structures, the rules and heuristics of procedural knowledge or the use of rules and equations to express relationships between variables symbolically.

This reductionist view of the knowledge maps neatly onto cognitive models of human knowledge processing, cognitive models of learning and cognitive approaches in artificial intelligence for representing knowledge, for example:

- (a) Cognitive Psychology's explanation of human problem solving as a process in which symbols refer to external phenomena, are stored in and retrieved from memory as propositional networks and manipulated and transformed by rules (Lindsay and Norman, 1977).
- (b) Cognitive teaching models from the field of instructional design, particularly instructional strategy models based on Gagné's conditions of learning paradigm.

Two such hierarchical approaches to instructional design are described by Wilson and Cole (1990):

"...learning hierarchies for analysing skills. A skill is rationally decomposed into parts and sub-parts, then instruction is ordered from simple subskills to the complete skill. Elaboration theory uses content structure (concept, procedure or principle) as the basis for organising and sequencing instruction. Both methods depend on

task analysis to breakdown the goals of instruction then as a method of sequencing that proceeds from simple to gradually more complex and complete tasks".

(p 49)

- (c) Cognitive approaches to knowledge representation for Artificial Intelligence (AI) such as semantic networks to express hierarchical inheritance (see Jackson 1990) production rules to convey heuristics, (Newell and Simon, 1972) and frames to encapsulate the properties, procedures and classification of objects in a single structure (Minsky 1975).

Rada (1991) has shown how these classical AI techniques could be used to provide hypertext (or hypermedia) systems with underlying expert system intelligence in order to represent some aspects of the implicit knowledge inherent in the connections between nodes, in addition to the explicit knowledge of the nodes themselves.

"By combining the techniques of expert systems with those of hypertext, one may build expertext systems which combine both the **intuitive power** of hypertext systems and the **formal power** of expert systems"

(Rada 1991 p 178)

To summarise, the Journeyman's view of the domain, the core knowledge, can be seen to follow a cognitive or symbolic paradigm, amenable to rational articulation and formal problem solving methods.

An objection to this approach of reducing the knowledge to a network of nodes and rules concerns the intrinsic validity of such an approach. Bereiter cites Harre's theory of the social nature of rationality which suggests that rules arise when people try to give retrospective explanations of their mental processes, which are instead justifications of their actions.

"What we call logical reasoning, and attribute to the workings of the individual mind, is actually a public reconstruction meant to legitimate a conclusion by showing that it can be derived by procedures recognised as valid."

(Bereiter, 1991 p14).

Bereiter (1991) argues further that rules are not an intrinsic part of human knowledge representation:

"However effective rules may be as an instructional device, the possibility remains that what is actually acquired is some complex, like a connectionist pattern, that merely approximates rule-based performance"

(p14).

Despite these reservations about symbolic approaches to the representation of knowledge it can be seen that they still have a contribution to make, for the following reasons:

- (a) They do provide a powerful shorthand for condensing a large amount of knowledge into a small space. To represent the same knowledge with say, many examples, would be extremely inefficient.
- (b) Part of the role of teaching should be "a public reconstruction meant to legitimate a conclusion." Without such a mechanism the validity of the knowledge itself is not accessible to debate. Intuitive understanding must be complemented by analytical understanding.
- (c) The problems for the implementors of the MLE, and its users, arising from the combinatorial explosion of knowledge may be containable. Many of the advantages of Rada's "expertext" approach to the design of an MLE could be retained if the explosion of cross references could be controlled. One method of achieving

this would be to subdivide the knowledge base into small, minimally-interacting segments. Connectivity within each segment could be sufficiently rich to express the key aspects of a concept without overwhelming the user. Connectivity between segments would be restricted to explicit "gateways." Users would have to make a conscious action in order to move between segments. This would enable them to "drop" the connections of the old segment (from working memory) before embarking on connections in the new segment.

5.2. Experiential or pattern-matching

Discussions with the Master during knowledge acquisition also revealed that domain experts apply their experience in ways which cannot easily be articulated or modelled, a type of knowledge which we described as "black magic". Examples of this involved situations such as planning flows from dams through easily erodable river basins for which there were no mathematical models or rules to explain hydraulic behaviour. Experts acquired appropriate knowledge of such situations by trial and error, either intentionally by using small scale physical models, or unintentionally by full scale design failures (from conversations with Professor G. Bullock, November 1992).

The need for a connectionist paradigm

The point here is that there appear to be certain types of knowledge within this domain which cannot be described by the reductionist, symbolic methods proposed by the Journeyman's perspective. Where it exists, knowledge of this type seems to take the form of pattern matching to previous experience.

Further investigation reveals other aspects of the problem domain, such as considerations of ease and cost of construction of concrete structures, which are likely to rely on such experiential knowledge. It is also likely that some aspects of the problem which can in theory be modelled by formal, rule-based approaches may in reality be solved by other methods. Carl Bereiter (1991) lists a number of examples to support this view:

- 1 Instead of being guided by rules of logic, people make use of mental models of situations, which they run in order to generate and test inferences (Johnson-Laird, 1983).

- 2 In dealing with probability and statistical inference, people rely on untrustworthy heuristics rather than formal principles. Easily remembered cases, for instance, have a major influence (Tversky and Kahneman, 1974).
- 3 Scientists, fully conversant with the formal laws of their domains, instead use informal, qualitative models in thinking about actual cases (Bobrow, 1985).
- 4 People's concepts, instead of being based on classification rules, are based more on family resemblances and prototypes (Rosch, 1978).
- 5 Metaphoric relationships play a much larger role in conceptual thought than has generally been recognised (Lakoff, 1987).
- 6 Self-reports of rule use are often unreliable; it can be shown that people's behaviour did not conform to the rules they said they were using (Nisbett & Ross, 1980).

(p 11)

Experienced Civil Engineers do not follow the prescriptions of formal models beyond the point at which they know that the resulting structures would not be cost-effective. For example, they may replace a theoretically optimal design for an elaborate spillway shape with a cruder shape which is cheaper to build. Discussions with engineers suggest that they use informal, qualitative models to make such decisions.

Laurillard (1991b) argues that "intuitive" understanding of scientific concepts is qualitatively different from "analytical" understanding:

"Students make use of a variety of aspects of their experience as they struggle to make sense of the language of an academic discipline: physical experiences, social experiences, emotions, intensional goals, irrelevant experiences - and these are important because they enrich the concepts they develop, and not always in beneficial ways. Against those powerful sensual experiences, academics put up the rather less compelling experiences of language, symbolism and analytical reasoning to develop students' conceptions of such concepts as 'velocity', 'power', 'structuralism', etc. Small wonder that for many students the sensual or emotional experiences hold sway, and the concept remains known "intuitively" rather than "analytically" as the academic would prefer".

(p 2)

Bereiter suggests that a connectionist paradigm is necessary to account for informal or intuitive levels of cognition. Briefly, this envisages knowledge as a vast network of interconnected elements in which the patterns of connections are significant, rather than the elements themselves. This view has a basis in neurophysiology and has been developed as a branch of AI in the form of neural networks. Bereiter provides an analogy for the approach, and it is more fully explained in Bechtel and Abrahamsen (1991). Bechtel and Abrahamsen also explain the resurgence of connectionism in the 1980's:

"a number of investigators began to confront the limitations of symbolic models. While initially the task of writing rule systems capable of accounting for human behaviour seemed tractable, intense pursuit of the endeavour raised doubts. Rule systems were hampered by their "brittleness", inflexibility, difficulty, learning from experience, inadequate generalisation, domain specificity and inefficiencies due to serial search through large systems. Human cognition which the rule systems were supposed to be modelling, seemed to be relatively free of such limitations."

(Bechtel & Abrahamsen, 1991. p17).

The Connectionist or pattern-matching paradigm is of value for considering experiential knowledge. This can be envisaged at two levels:

- (i) Visual pattern matching of dynamic phenomena, such as the erosion of stream beds, the force of large, fast water flows, and "the

"hydraulic jump" (a standing wave caused by changes of speed of fluid flow) for example Figure 14.

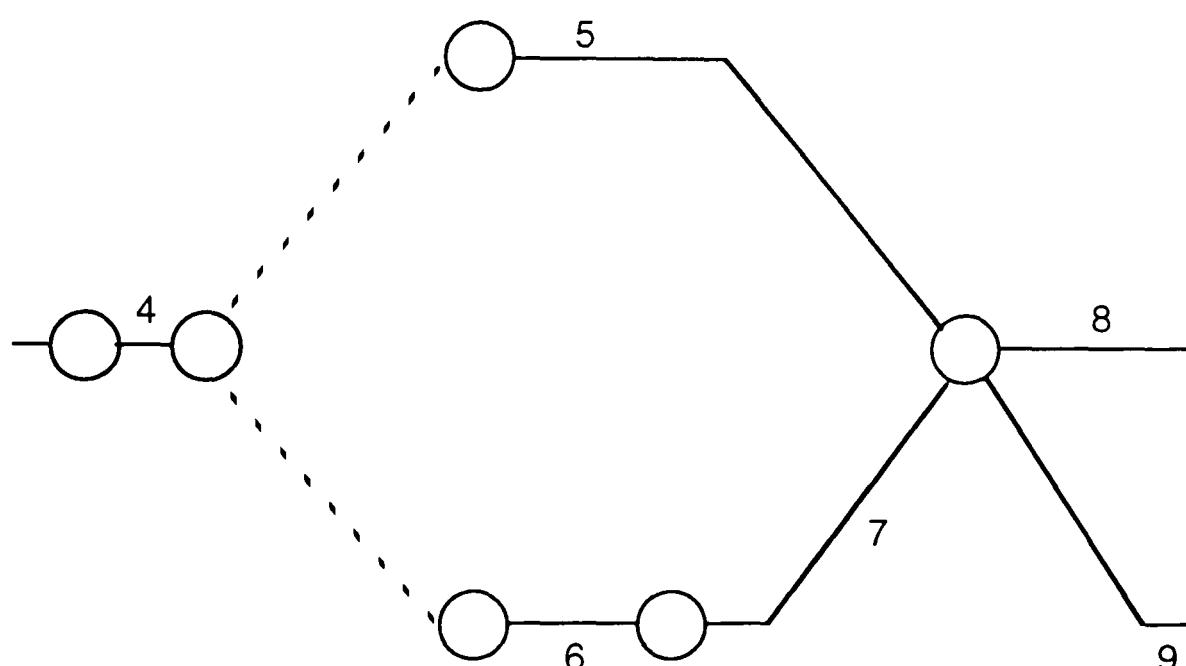
Figure 14. "Stills" of a hydraulic jump taken from a video filmed in a laboratory by the author.



Such dynamic phenomena can be described in words or diagrams, but can really only be expressed vividly with the benefit of motion video and animation. Abstract or conceptual dynamic systems, for example, triangles of forces, also fall into this category, and would become more understandable through dynamic presentation in animation, or video of real world phenomena. Laurillard (1991b) provides evidence of the value of simulation in computer-based learning, and guidelines for the ways in which they should be used. If used correctly simulations can reinforce both intuitive and cognitive understanding. Juxtaposing simulation with symbolic representation helps to make the acquisition of difficult concepts easier while reducing the risk that intuitive "understanding" may be based on misconceptions.

- (ii) Temporal or meta-cognitive pattern matching. There are important meta-level patterns in this and similar domains which are based on long time frames in the real world; for example, the idea of a project structure with clearly definable phases, activities, durations, deliverables and dependencies, extending over periods of months or years expressible graphically as an activity network (see Figure 15).

Figure 15. An example of explicit metacognitive knowledge: a PERT network



No.	Activity	Earliest Start	Latest
..
4.	Obtain Authority to Proceed	1/3/94	
5.	Hydrological Survey	2/3/94	
6.	Geological Survey	2/3/94	
7.	Prepare Access	7/4/94	
8.	Excavate Footings	8/10/94	
9.	Excavate Tunnel 1	8/10/94	
..

Exactly what the principles of a connectionist approach to instructional design are, is not made clear in the literature that has been reviewed. It has been seen that there are flaws in a rule based approach and that, at an intuitive level, a

connectionist model of learning is appealing for some sorts of knowledge. However, none of the clear strategies and techniques which are available for symbolic approaches, for example Vassileva's, (1991) pedagogic structure of the domain have been found.

As a working strategy the following principles were adopted:

- (a) Where phenomena cannot be explained by rules they should be portrayed as clearly as possible using multimedia techniques. The characteristic features of the phenomena, for example physical changes or temporal changes should be identified to enable the learner to recognise them in real world problem solving. Heuristics may be appropriate to articulate this characterisation.
- (b) Where phenomena can be explained by rules they should also be simulated where intuitive understanding is hard to achieve. The characteristic features of the phenomenon should be related to the symbolic explanations, for example, changes in flow behaviour being related to changes in the values of variables in an equation. Some principles may be hard to grasp in the form of rules on their own, even though rules may subsequently be a convenient notation for discussing or applying the principles.
- (c) The way in which simulations are presented, to students, manipulated by them and integrated into a learning environment should be carefully planned and executed within some kind of instructional framework, for example, Laurillard's (1991b) guidelines. These cover the preparation of the students and the goals they are set, the need for demanding open-ended tasks in the students' interaction with simulations and the need for discussion

and reflection on what has been experienced.

There can be other good reasons for using interactive animation s and video. For example, it is good for training in psychomotor skills and it can be engaging. However, it can also be expensive to produce. The working principles above do at least provide some rationale for deciding when to use the more expensive techniques and when they are not necessary.

5.3. Metacognitive or task-model-based

Metacognitive or task-model-based knowledge provides the structure within which the engineer can apply Symbolic and Experiential knowledge to perform useful work. The most obvious form of this knowledge, which was elicited from industry representatives, is the life-cycle knowledge of expected project phases, activities and deliverables that would be required by a company for a particular project.

As well as this high level, "long time scale" knowledge used to manage a project, there is also the lower level "shorter time scale" knowledge of strategies and heuristics that are needed to order and control the completion of individual project life-cycle tasks. Such tasks may include, for example, design work, report writing, presentation giving or participation in meetings. Many of the strategies and heuristics that are needed to complete a typical design task were elicited from the journeyman.

The methods used to structure and control whole projects and the techniques needed to accomplish individual tasks can be taught as sets of rules. However, this knowledge is most usually, and probably most effectively, taught through repeated exposure to project work. So, the best way to convey this knowledge (and provide a framework for the understanding and application of Symbolic and Experiential knowledge) would seem to be to simulate in some way, on computer, some aspects of real life project work.

5.4. Summary

5.4.1. Knowledge representation paradigms

Analysis of the acquired knowledge determined that it could be categorised under three broad classes of representation paradigm. Each of these paradigms is summarised below together with the particular types of knowledge that are identified with each of them.

- (1) **Symbolic** (elicited from the “Journeyman” and confirmed by the “Master”) i.e. conceptual, factual and procedural knowledge.
 - (a) The conceptual and factual knowledge of natural phenomena and engineered structures which can be effectively expressed as equations, graphs or rules to show the relationships between the values involved.
 - (b) The procedural knowledge of, for instance, the steps needed to solve an equation or the steps involved in deriving an equation.
- (2) **Experiential** i.e. the natural patterns of past experience which are built up and refined automatically through repeated exposure to phenomena and which improve the understanding of a reoccurrence of the phenomenon. The two types of Experiential knowledge identified were:
 - (a) Knowledge and intuitive understanding of dynamic systems such as complex real-world physical phenomena which can only be fully understood through some physical sensory process e.g. the erodability of a stream bed. Such complex phenomena are difficult to understand through symbolic means alone because the symbolic methods needed to represent them with any degree of accuracy are indigestibly complicated.

- (b) Intuitive understanding of values, relationships and abstract theoretical constructs (which may not be observable in the real world) which can be expressed symbolically but which are hard to grasp unless they can be sensorily experienced e.g. the force of 1,000 cubic meters of water flowing down a spillway; the kinetic energy of water before and after an hydraulic jump; the triangle of forces.

Type (a) was disclosed by the “Master”, type (b) stressed by both the “Master” and the “Journeyman”.

- (3) **Metacognitive** i.e. the framework within which the Symbolic and Experiential knowledge is applied in order to accomplish a task. Knowledge of how to heuristically select, adapt and apply a framework to a particular context is also needed. In civil engineering the levels of knowledge needed within such a framework include:

- (a) High level project structuring metacognitive knowledge i.e.: knowledge that enables the engineer to select and order tasks so as to produce an appropriate, workable, coherent and comprehensive, project structure of a long time scale e.g. 3 months. This includes knowledge of the typical Civil Engineering project life-cycle, its phases, activities and deliverables and how the project life-cycle is adapted according to individual company practice and the needs of the particular project being undertaken. The importance of this type of knowledge was emphasised by industry practitioners.
- (b) Lower level, applicable over a shorter time frame (e.g. 1 day), knowledge of strategies and heuristics which are needed to be able to apply the symbolic and experiential knowledge so that an individual task within the project life-cycle can be accomplished. Typical tasks within the project life-cycle include: solitary, technical

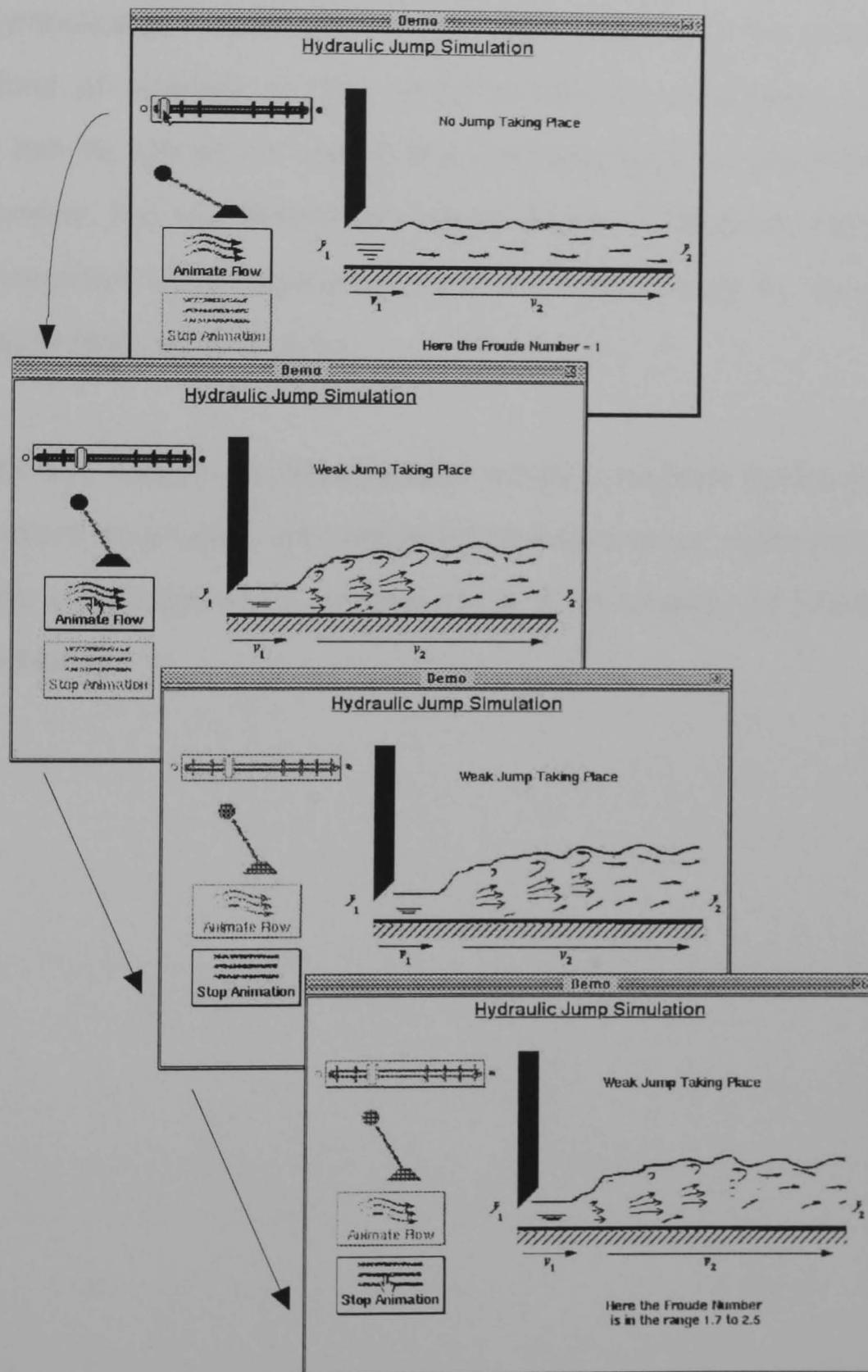
design tasks and also tasks used to communicate, present and discuss designs with superiors, sponsors and other members of the design team. The strategies needed to perform a design task were elicited from the "Journeyman" during his solution of the example design problem.

5.4.2. Implied most suitable presentation styles

Each category of knowledge can be seen to be most suited to particular methods of presentation:

- (a) **Symbolic knowledge** e.g. mathematical relationships between flow rates and spillway dimensions, needs to be expressed in symbolic terms, i.e. equations or rules. However, much symbolic knowledge can often also be expressed using experiential techniques such as simulation, animation, video and pictures so that it becomes understandable at an intuitive level through sensory experience. An example from a simulation is given in Figure 16.

Figure 16. "Stills" taken from an animated simulation of the Hydraulic Jump created by the author.



(The hydraulic jump phenomenon is used within hydraulic jump stilling basins to dissipate energy and convert the shallow fast flowing water leaving the base of a spillway into deeper, slower moving water which more closely matches the river below the dam).

- (b) **Metacognitive knowledge** is often learnt through experience, e.g. the patterns of procedures and approaches which best suit particular real world problems. Although it is possible to teach such strategies and heuristics as rules (i.e. symbolically) in reality, a thorough understanding of the subtleties and variations of strategy is best acquired through experience. Such experience can be offered on computer by simulating real world problem solving. However, it is also beneficial (Collins, Brown, & Newman, 1989) to make metacognitive knowledge explicit with such techniques as life-cycle models, activity planning and so on.
- (c) **Experiential knowledge** can thus be seen not as a separate category but as an important alternative, connectionist perspective on symbolic and metacognitive knowledge which enables these to be understood intuitively as well as formally.

6. Pedagogic models: the requirement for a pedagogic model

To be able to present to the learner in an effective way the types of knowledge identified in chapters 4 and 5, a coherent and comprehensive framework is required which is rich enough to include the various types of knowledge, yet structured clearly enough to make their presentation digestible and also adaptable enough to be applicable to the design of a computer based delivery system. There are many teaching models in existence which might possibly be useful and so need to be considered. Wilson and Cole (1991) distinguish between two broad categories of model, those which have emerged from instructional design theory and those which have emerged from cognitive psychology. This chapter discusses the teaching models that have emerged from each of these categories to help determine what is needed in the pedagogic strategy to be used by the MLE.

6.1. Models based on Instructional-Design Theory

Wilson and Cole (1991) assert that all models emerging from instructional-design theory that offer instructional strategies are based on Robert Gagné's "conditions-of-learning paradigm" (Gagné, 1966), which was they say in its time a significant departure from the dominant Skinnerian operant conditioning (more information on Skinnerian operant conditioning is given chapter 3, 'Section 3.1.1. Drill and Practice'). The conditions-of-learning paradigm presumes that the learning of, say, a rule, a psychomotor skill, an attitude, or a piece of verbal information can be accomplished by using task analysis to decompose the instructional goal into a graded hierarchy of behaviourally specific learning objectives. The objectives can then be classified according to the types of learning they involve and so

matched to suitable instructional strategies. The sequencing of instruction then proceeds from simple to gradually more complex and complete objectives. For instance, Gagné produced a method for analysing skills to produce "learning hierarchies". With this method a skill is rationally broken down into parts and sub-parts, then instruction is ordered from simple sub-skills to the complete skills.

Conditions-of-learning models for decomposing and sequencing learning have been used within a number of computer based systems, for example, Vassileva's (1991) intelligent tutoring system (ITS) which teaches the laws of electric current. Vassileva decomposes the knowledge to be learnt into a "pedagogical structure of the domain" (PSD). This PSD consists of the concepts (laws) which are to be taught (for example, the Ohm law) and the order in which they can be acquired. The belief is that to understand a complicated concept, a number of prerequisite, underlying concepts first need to be understood. The PSD can be represented as a directed graph with nodes which correspond to the concepts. The nodes are then connected according to precedence rules. Each concept has one or more teaching strategies associated with it (for instance, a mathematical definition and a simulated experiment are associated with Ohm's law). The choice of strategy used to teach a concept depends on certain criteria, such as, which strategy proved successful with the student in the past. The ITS generates a number of paths through the graph, from the concepts which are deemed to already be known by the student to the concept which represents the session's learning objective. A path is then chosen so the student is taught each concept in turn according to the strategies selected. The choice of path is again determined by some criteria, for example, the shortest route. Additional facilities are provided to enable such things as the updating of a student model and the switching to an alternative teaching path if the teaching reaches an impasse. The main focus of this approach is to teach a structural model of the concepts of a domain, abstracted from any practical context.

The influence of the conditions-of-learning paradigm can clearly be seen within a number of other traditional instructional-design theories, for example, the Component Display Theory (Merrill, 1983) and the Elaboration Theory (Reigeluth and Stein, 1983). All these models remove the knowledge and skills to be learnt from the context of their application. Such models might be useful for teaching elements of the “Symbolic” knowledge identified and discussed in chapters 4 and 5, however they underplay the importance of the role of “metacognitive” knowledge and the value of “experiential” knowledge. The Civil Engineering graduate trainee needs to know how the abstract, theoretical symbolic and factual knowledge (which is usually stressed during their undergraduate training) relates to real-world problem contexts, both physically and in terms of when and how it should be applied. In fact, in recent years, researchers within a number of fields have began to question the assumption that knowledge can exist independently of its context (Resnick, 1989).

6.2. Models developed from Cognitive Psychology and Constructivism

Wilson and Cole (1991) note that a number of cognitive teaching models emphasise design elements that traditional instructional-design models have historically under emphasised. They suggest, for example, that constructivist theorists have offered alternatives to the conditions-of-learning paradigm and that many cognitive teaching models share a number of similar constructivist concepts. Somekh (1994) sees the main debate in the 20th century amongst those involved with thinking about learning as being between:

“those following Skinner, who see learning as “conditioning” by means of a carefully planned sequence of “stimuli” designed to elicit a set of desired “responses”; and those following the ideas of Vygotsky and Bruner, who see knowledge as “a process rather than a product” (Bruner, 1966, p72), constructed by the learner as a result of

experiences, critical reflection upon those experiences, and social interaction (e.g. discussion) (see Prawat, 1991)."

(Somekh, 1994, pp2-3)

Somekh (1994) cites Desforges (1989) as rejecting the confrontational stance between behaviourists and constructivists and proposing that there is a need for both kinds of learning. According to Desforges there are occasions when seemingly arbitrary facts need to be learnt and other instances when concepts need to be learnt. There is usually a link between the two types of knowledge. However, each can require a quite different learning process, as Desforges says:

"The latter is knowledge constructed by human intelligence in interaction with and adaptation to the environment and is unlikely ever to be understood unless reconstructed by the learner."

(Desforges, 1989, p20)

For a particular learning task, developers of courseware have to strike an appropriate balance between attempting to facilitate behaviourist and the far more difficult and costly constructivist learning.

To help support constructive learning in courseware, Somekh (1994) suggests a number of different strategies relating to:

Mental schemata

Central to the theory of constructivist learning is the notion that each individual develops their own mental schemata which can later be used by the individual to help them operate with confidence in a complex world.

This theory has two implications for courseware development:

1. To learn, individuals have to be able to build on their existing schemata.

This suggests that learners should have control over the pace and direction of their learning.

2. To learn, individuals need support to build new schemata

This implies the necessity for an underlying structure to the courseware. As an example, Somekh (1994) suggests how a book can provide such a structure through its table of contents, partition into chapters and use of headings and sub-headings and so on.

In courseware, to ensure the underlying structure of the material is not obscured, the freedom of movement given to the learner may have to be limited. One way of doing this might be to give learners some small-scale, local control within a larger, overall guiding structure.

To build schemata, learners also have to be given the capability to revise and revisit, perhaps many times, any material they are unsure of. Somekh (1994) cites Bruner (1966) as addressing this need in his “spiral curriculum” in which learners are presented with a succession of concepts, each in a variety of ways, over a period of time (specifically through different kinds of content involving slightly different approaches).

Situated learning

Recently many researchers have stressed the need for learning to be “situated” to enable schema development (Brown et al., 1989). From this author’s research the need for such “situated” learning can be seen to correspond to the need elicited by this author for realistic “experiential” knowledge of “symbolic” knowledge and “metacognitive” skills (see ‘Chapter 4. Knowledge acquisition’ and ‘Chapter 5. Models of human knowledge: analysis of the information gained

from knowledge acquisition').

A powerful characteristic of "situated learning" is the capability of learning from expert colleagues who model appropriate behaviours and approaches to problems. Somekh (1994) suggested a good way of doing this in courseware might be to use examples of problem-solution by posing problems and offering the learner the option of being given step-by-step solutions to the problems with explanations. However, she warns that if such an approach is used there is a danger of the learner simply learning formulaic responses instead of actually understanding the concepts involved.

Critical self-reflection

Constructivists stress that effective learning requires learners to actively work with and question ideas so they can construct their own knowledge and so develop appropriate mental schema. To help achieve this Somekh (1994) cites De Corte (1990, p73) as seeing a role in educational software for:

"enhancing the acquisition of meta-cognitive skills and learning strategies through the explication of, and reflection on (the learner's) knowledge (deficiencies and misconceptions versus strengths) as well as on their thinking methods and learning activities (powerful versus weak)."

Ways of realising the above in educational software, suggested by Somekh (1994), include offering an on-screen note-book for recording self-reflections. Somekh (1994) proposes that this can be extremely effective if the learner is later required to use the note-book in discussions with peers or tutors.

Motivation

Somekh (1994) deems motivation to be essential to learning. One problem which she believes may reduce a learner's motivation to use a piece of courseware is if the learner has had past negative experiences of using a computer or if the learner perceives themselves to be a non-technology person. She suggests how producers of educational software can attempt to overcome this problem by developing environments which do not convey a hard technological image.

“Flow” or cognitive engagement

According to Somekh (1994), uninterrupted presentations can induce a high level of concentration in learners which has been termed “flow” by Csikszentmihalyi, 1982 or alternately, “cognitive engagement” by Kozma, 1991. Somekh (1994) suggests that courseware is unlikely to induce “flow” if the learner’s thoughts are interrupted too frequently by a program requiring answers or providing unsolicited information.

Translation between symbol systems

Somekh (1994) cites Bruner’s (1966) concern that one of the major problems in trying to learn is the necessity to “translate” concepts from a symbol system such as language, numbers or graphical representation into meaningful mental schema. For example, to understand concepts presented in a textual explanation learners have to translate the explanation in terms of their own experience to be able to link it to their existing mental schema. Content screens can make heavy demands of this kind on the learner. Even if graphics are included the corresponding need to reduce the number of words on the screen can lead to

truncated text which is unable to adequately explain the concepts involved. One possible solution suggested by Somekh (1994) is to provide separate printed materials to be used together with the educational software.

For learners, the difficulty of “translation” also encompasses the problem of “transforming” meaning between symbol systems. One such common problem involves interpreting graphs that are accompanied by explanatory text (Mokros and Tinker, 1987; Smith, 1993). Somekh (1994) believes there is a tendency for tutors to underestimate the problems novices have in interpreting graphs and it is wrong to assume graphical representations will help explain text when often this approach simply overlays a second level of difficulty.

One way suggested by Somekh (1994) to help learners transform meaning between text and graphical symbol systems would be to actually provide on a computer a dynamic model of the transformation between text and graph (Kozma, 1991). This would involve the user changing the text and the computer “transforming” this into changes in the graph. In this author’s research such an approach corresponds to the elicited need to “experience” the meaning of much “symbolic” knowledge (see chapters 4 and 5) and in particular, such an approach is put forward as being appropriate in ‘Section 5.4.2. Implied most suitable presentation styles’.

Feed-back and assessment

Major obstacles invariably occur when trying to satisfactorily assess learning since cognitive processes are private, hidden and unique to the individual. These problems are made worse when attempting to incorporate assessment within an educational software package because in such a situation learners would usually be unable to simply reply to questions using natural language as they

would to a human tutor. Yet, some kind of feedback or self-assessment seems vitally important to the learning process.

With computer controlled assessment, Somekh (1994) believes it is difficult to prevent testing from stressing the lowest A and B levels of the MacDonald (MacDonald et al, 1976) typology of interactions (A: recognition, B: recall, C: reconstructive understanding or comprehension, D: global reconstructive or 'intuitive' understanding, E: constructive understanding) - at the expense of even level C interactions, simply because the former are so much easier to test than the latter. Somekh (1994) feels attempting to incorporate into courseware sophisticated assessment methods or mechanisms for 'tracking' learners' progress can place too high a constraint on the educational design of the software. One alternative put forward by Somekh (1994) is to produce supplementary, paper-based assessment materials to be used after completion of a module of courseware.

Laurillard (1993) also discusses some recommendations for constructivist instructional design which were presented in the May and September, 1991 issues of Educational Technology and the resulting book by Duffy & Jonassen (1992). In Duffy & Jonassen (1992), many of the authors believe that purposeful knowledge construction may be facilitated by learning environments which:

- Provide multiple representations of reality, thereby
- Avoiding over simplification of instruction by representing the natural complexity of the real world,
- Focus on knowledge construction, not reproduction,
- Present authentic tasks (contextualising rather than abstracting instruction)
- Provide real-world, case-based learning environments, rather than pre-determined instructional sequences,
- Foster reflective practice,
- Enable context- and content-dependent knowledge construction, and

- Supports collaborative construction of knowledge through social negotiation, not competition among learners for recognition.

It seems clear that much of the elicited “symbolic” knowledge (see chapters 4 and 5) could be effectively conveyed by behaviourist means. However to effectively, “experientially” convey the elicited “metacognitive” knowledge and the remaining “symbolic” knowledge would require the use of constructivist techniques.

The preceding review of constructivist learning requirements for a pedagogic model imposes a demanding brief for the design of the MLE. In fact a highly suitable model exists in the Cognitive Apprenticeship Model (CAM) of Collins, Brown and Newman (1989). The CAM can be seen to address most of the important cognitive issues for pedagogic models identified earlier in this section and to map well onto the domain knowledge described in chapters 4 and 5. It was not developed to be applicable to only one specific domain or to only one particular instructional format, that is, it aims to be able to be usefully applied to varying domains and to individual instruction, group instruction or even computer-aided instruction. However, each feature of the model had to be re-interpreted to determine how it could be specifically applied to inform instructional design for the domain of Hydraulic Civil Engineering and the instructional format of using computer based multimedia as the delivery system.

Each of the elements of the cognitive apprenticeship model is reconsidered in the following chapter in relation to the types of information in our domain which it can best represent. Also considered is how the MLE should be designed to accommodate these types of information and the cognitive apprenticeship model.

7. The instructional basis of the MLE

7.1. An overview of the Cognitive Apprenticeship Model (CAM)

The Cognitive Apprenticeship Model (CAM) developed in Collins, Brown and Newman (1989) and Collins (1991) is an adaptation of the traditional approach of craft apprenticeship for skill learning into methods for teaching and learning the reasoning and problem solving cognitive skills of a domain. They have produced a general framework (which goes beyond the techniques of traditional apprenticeship) for the design of learning environments. In their terms, "environment" includes the content taught, the pedagogical methods employed, the sequencing of learning activities, and the social components of learning.

Below is a summary of the characteristic features of the CAM, as discussed in their papers and the paper by Wilson and Cole (1991). In Collins, Brown and Newman (1989) the general CAM framework was developed from an examination of three pedagogical "success models" that exhibited elements of apprenticeship in teaching the thinking and reasoning skills needed within the domains of reading (Palincsar and Brown's Reciprocal Teaching of Reading, 1984), writing (Scardamalia and Bereiter's Procedural Facilitation of Writing, 1985; Scardamalia, Bereiter, and Steinbach, 1984) and mathematics (Schoenfeld's Method for Teaching Mathematical Problem Solving, 1983, 1985). In the discussion below, where appropriate, examples provided in Collins, Brown and Newman (1989) from one or more of these domains will be used to help clarify the explanation of the model. A detailed discussion of the representation of civil engineering knowledge in the CAM follows in section 7.2.

The Cognitive Apprenticeship model:

This is described by Collins, Brown and Newman as having eight major components.

1. **Content:** heuristic knowledge and textbook knowledge.

Collins et al. (1989) refer to four types of knowledge:

- 1.1. **Domain knowledge:** conceptual, factual and procedural knowledge usually found in textbooks.

Important, but often, insufficient to enable students to approach and solve problems independently. In mathematics, domain knowledge includes number facts, definitions and the procedures for solving different kinds of problems. The procedures range from simple addition algorithms to more complex procedures for solving algebra problems and constructing proofs in geometry.

- 1.2. **Heuristic strategies:** "rules of thumb" which speed up problem solving.

Experts normally acquire heuristic knowledge as a side effect of repeated problem solving. Unfortunately, slow learners generally fail to gain this subtle knowledge. However, there is some evidence that at least some heuristic knowledge can be made explicit and represented in a teachable form (Chi, Glaser, and Farr, 1988). For example, writing the introduction to a text after the text has been written is a widely used writing heuristic. It is a popular tactic because experienced writer's recognise that their

final draft of a text will usually deviate significantly from their initial plan.

- 1.3. **Control strategies:** the knowledge needed to manage the carrying out of a task.

While carrying out a task, the learner must be able to select, from perhaps many possibilities, the most suitable strategy to apply next. Differing activities within the overall task may require different strategies. A strategy which was thought to be suitable may prove to be ineffective and a different strategy will need to be applied.

Different levels of control strategy can be identified. For instance, there are control strategies which can manage problem solving at a global level and are applicable within a number of domains. For example, a basic control strategy for solving a complicated problem would have the solver who had reached an impasse begin work on a new part of the problem. Alternatively, there are lower level control strategies concerning the selection of domain-specific problem-solving heuristics and strategies which enable a particular part of the current task to be carried out.

Each control strategy has a **monitoring**, **diagnostic**, and **remedial** element, for example, deciding how to proceed in a task usually depends on

- (a) assessing how the current state of the problem compares to the goals of the problem - **monitoring**. e.g. For reading, where the goal is understanding, Baker and Brown (1980) and Collins and Smith (1982) have identified comprehension monitoring strategies, such as, attempting to give the main point of a paragraph which has just been

read. If this cannot be done then the paragraph has not been understood. Monitoring strategies lead either to diagnosis strategies or directly to remedial strategies.

- (b) analysing current difficulties - **diagnosis**. e.g. For reading, if a paragraph has not been understood (as determined by a monitoring strategy) then Palinscar and Brown (1984) suggest a strategy of clarifying difficulties with text. Here the reader attempts to identify the particular word or phrase which they cannot understand.
- (c) the strategies available for dealing with difficulties - **remedial actions**. e.g. For reading, having realised they do not understand a paragraph (by using a monitoring strategy) the reader may simply reread the paragraph. Alternatively, if they have used a diagnosis strategy and identified the words or phrases they cannot understand they may wish to look up those words in a dictionary.

1.4. **Learning strategies:** strategies for learning any of the above types of knowledge.

These can range from general strategies for understanding a new domain to more local strategies for gaining or adjusting the knowledge that is needed to continue solving a problem or carrying out a complex task. e.g. To improve their reading, students have to be able to select books that increase their vocabulary but which they still find readable. They also need to be able to check their understanding by reading reviews or through engaging in discussion. To improve, writers have to know how to find people who can offer constructive, reasoned critiques of their work. They, in turn, need to be able to analyse the work of others.

Maths problem solvers, using text books, need to realise that they will gain more benefit from worked examples if they compare their own solutions to them rather than examine them alone.

2. Situated Learning: Teaching knowledge and skills in the contexts in which they will be used in real life, hence the idea of "apprenticeship."

Brown, Collins and Duguid (1989) stress the need to place all learning within "authentic" contexts that imitate real-life problem-solving situations. Collins (1991) is not so insistent on the importance of constructing real-life settings but he does emphasise the need for placing instruction within a problem-solving context. He cites a number of benefits to be gained from doing this, including

- Learners learn to apply their knowledge under appropriate conditions.
- Learners will realise the implications of new knowledge. Student learning in classrooms can often be seen to suffer, through lack of motivation, because the students fail to see the relevance to their lives of the material being learnt. Acquiring knowledge in order to solve meaningful problems can provide immediate proof of the usefulness and importance of the knowledge.
- Knowledge will be put into memory in a way that makes it easily retrievable when solving similar problems. People tend to retrieve knowledge more easily when in the same setting as its acquisition.

The three teaching models from the domains of reading, writing and mathematics considered by Collins, Brown and Newman (1989) all place learning within the context of "problem-solving" or the completion of a meaningful task. Collins (1991) suggests that some aspects of real-life can

be brought into, for example, mathematics teaching by using settings "ranging from running a bank or shopping in a grocery store to inventing new theorems or finding proofs. That is, situated learning can incorporate situations from everyday life to the most theoretical endeavors" (p. 122).

3. **Modelling and explaining:** presenting the skills as part of an overarching process and giving reasons for the way it evolves.

Collins (1991) lists two kinds of modelling:

- (1) Modelling of real word processes. For example, Collins (1991) cites showing the movement of electrons in circuits in Haertel (1987) as being an example of this. Another example from Collins (1991) is the illustration of how information coded in DNA is translated into protein molecules.
- (2) Modelling of expert performance, to give the learner a view of the correct way to do things, including hidden cognitive processes. For example, teachers can model the reading process by reading aloud in one voice, while verbalising their thought processes (e.g. formulating and trying out hypotheses about the meaning of the text, the authors intentions and what they believe will happen next) in a different voice (Collins and Smith, 1982).

Both of these processes can be modelled with the aid of computers. Collins emphasises the integration of both the demonstration and the explanation during instruction.

As they observe the modelled performance, learners need to be able to view

related explanations. Computers are well suited to making explicit hidden processes that would otherwise be difficult to observe. Collins suggests that fully modelling competent performance, i.e. by including any false starts, dead ends, and remedial strategies, would help learners to adopt more quickly the implied forms of knowledge discussed in section (1) Content.

In this way, teachers are seen as "intelligent novices" (Bransford et al., 1988). By viewing both the process being modelled and the corresponding explanations, students can develop "conditionalized" knowledge, i.e., knowledge about when and where knowledge should be applied to solve various problems.

4. Coaching, including Scaffolding and Fading.

4.1. Coaching involves monitoring learners as they work and offering hints, scaffolding, feedback, modelling, reminders, and new tasks all of which are intended to move their performance closer to the performance of an expert. Coaching can direct attention to unnoticed or forgotten aspects of the task. Coaching stresses the absorption and integration of skills that are used while working towards a well understood goal by providing highly interactive and highly situated feedback and suggestions. Exactly when and how to incorporate coaching into instruction, especially as related to learner errors and misconceptions, continues to be an issue. For example, should an attempt be made to prevent learners from making mistakes or should learners be allowed to fail, perhaps even required to learn from failure, as in Clancey's (1986) medical tutor? While reading, the task might be to construct summaries of various texts, perhaps, in an attempt to demonstrate understanding. The teacher (coach) could

choose texts which present particular difficulties, remind the student that a summary should synthesise the whole text into one or two sentences, suggest how to begin to construct a summary and then evaluate the summary produced by the student.

4.2. Scaffolding refers to the supports that are supplied to assist the learner through the completion of a task. When scaffolding is provided by a trainer, the trainer carries out parts of the overall task that cannot yet be managed by the trainee. It is intended that the trainee takes on as much of the task as possible, as soon as possible, until they can comfortably perform the whole task on their own. To accomplish this there needs to be an accurate diagnosis of the trainee's current level of proficiency or current difficulty and the existence, within the task, of an intermediate step at the appropriate level of difficulty. Typical supports can include prompts or procedural facilitations such as those provided on cue cards by Scardamalia et al (1984) e.g. "A better argument would be..., A different aspect would be..." which are offered during their "generating a new idea" stage of writing planning. Palincsar and Brown's (1984) Reciprocal Teaching provides prompt like suggestions and help to support their students reading tasks.

4.3. Fading involves the progressive removal of supports until the trainees are working on their own.

5. Articulation.

This involves requiring learners to think about their actions and provide reasons for their decisions and strategies in order to make their tacit

knowledge more explicit. Collins (1991) cites the benefits of this as added insight and the ability to compare knowledge across contexts. As learners' tacit knowledge is brought to light, that knowledge can then be used when solving other problems. Wilson and Cole (1991) cite think-aloud protocols as one example of articulation (Hayes and Flower, 1980; Smith and Wedman, 1988). This is used in Scardamalia et al's (1984) method of Procedural Facilitation of Writing. Inquiry teaching (Collins and Stevens, 1982, 1983) offers another example of articulation. Inquiry teaching as applied to reading might involve the teacher systematically questioning students as to why one summary of a text is good but another poor, thereby, compelling the students to construct an explicit model of a good summary (Collins et al 1989).

6. Reflection.

This requires learners to reexamine their attempt to finish a task and analyse their performance. Reflection can be thought of as articulation that is pointed backwards to past experience. In Brown, (1985 [a,b]) and Collins and Brown, (1988) learners can compare their own problem-solving processes with those of an expert or with those of other learners. Schoenfeld's (1983, 1985) method for teaching mathematical problem solving uses a form of reflection performed by both the expert and the student which Schoenfeld calls postmortem analysis. For a given problem the expert models the problem solving process then retraces the solution method, drawing attention to generalisable features. For example, the expert may recall any heuristics used, points where alternatives were generated, and why a particular alternative was selected, and so on. Students also perform the same sort of postmortem analysis on their homework in front of the class with the rest of the class then critiquing their solution method. This allows students to compare expert problem solving processes with student problem solving

processes which should then help them identify and correct any common erroneous aspects of student performance which they may already or would possibly have gone on to exhibit.

Reflection can be enhanced by the use of various techniques, e.g. computers, video or audio-tape recorders for reproducing or "replaying," for comparison, the performances of both the expert and the novice. Replays may vary in the detail they provide according to, perhaps, the learner's stage of learning. Although, usually, some form of "abstracted replay" (Collins et al 1989) is used where the determining features of expert and novice performance are highlighted (as in Schoenfeld's postmortem analysis, described above). To help with reading or writing, a tape recorder may be used to record both experts and novices think aloud as they perform. The tape can then be replayed later for comparison.

Without reflection, learners may not learn: (i) how to recognise the appropriate situation for applying their knowledge, (ii) which particular procedure should be used or (iii) how their knowledge can be transferred to different tasks.

7. Exploration.

This stage allows students to try out different strategies and hypotheses and examine their consequences. Collins (1991) suggests that exploration helps students learn how to seek knowledge autonomously and to set and try out hypotheses. It requires fading in problem-setting as well as the usual fading in problem-solving. Exploration could be taught by setting students general goals and then encouraging them to work on specific subgoals which they are interested in. Students may even be allowed to alter goals if they discover something more interesting to pursue. For example, in reading , the

teacher might send the students to the library to find out which president died in office as a result of a trip to Alaska or to investigate theories about why the stock market crashed in 1929 (Collins et al, 1989).

8. Sequence.

This concerns the need to order instruction from simple to complex, with increasing diversity, and to present global skills before local skills.

- **Increasing complexity.** Collins et al. (1989) achieve this by assigning increasingly complex tasks or, within a task, by increasing the fading of the scaffolding being used, i.e. fading the group or trainer support for individual problem solving. The complexity of tasks within most domains can usually be seen to vary along a number of dimensions. For example, in reading, the complexity of texts can vary according to vocabulary, syntax, level of conceptual abstraction and difficulty of argument. Increasing complexity could involve moving from short texts with simple syntax and vocabulary and which use natural descriptions to difficult texts using complex, abstract ideas.
- **Increasing diversity** requires variation of the tasks and task situations so the student can learn when and when not to apply skills and strategies. This could be done by providing a sequence of tasks in which a wider and wider variety of strategies or skills are required. Seeing how knowledge can be used in a number of circumstances should make the knowledge more readily available for application to novel situations. Increasing diversity in writing could be achieved by setting varying problems such as writing descriptive or instructional text as opposed to writing to convince an audience of your point of view. Alternatively, the problem situation could be varied by, for instance, incorporating particular constraints, such as, writing an essay within a

given time limit or writing for a particular audience (say the school board).

- **Global before local skills** requires the support offered to learners to relieve them of the lower level skills that are needed to perform a complex task so that they first examine, practise and learn the higher level reasoning skills necessary to carry out the task. This allows learners to develop a "conceptual map" of the overall activity before they go on to develop specific skills. In algebra, for example, students may be relieved of having to carry out low-level computations where they as yet lack skill so that they can initially concentrate on the higher order reasoning and strategies required to solve an interesting problem, Brown (1985 b). Building up an early conceptual model of the task (which can also be accomplished through expert modelling and explanation as discussed in section 2. Modelling and explaining) provides a number of benefits. For instance, even when only a small part of the task can be carried out by the learner, having an overall conceptual model of the activity allows the learner both to understand the segment they are currently working on and have a clear goal to aim for as they link together an increasing number of task segments. Also, an explicit conceptual model of the goal task will act as a guide for the learner that should help them develop their self-monitoring and self-correction skills. Collins et al (1989) also suggest that having such a model should crucially prevent students from gaining misconceptions about the applicability and relationship to other processes of individual skills.

Collins et al. (1989) express the view that the framework will be useful to the field in studying, designing, and evaluating pedagogical methods, materials, and technologies. They suggest that the core techniques of modelling, coaching and

fading can be formalised and embedded in future computer programs. They also suggest that computers make it possible to give more personal attention to individual students, without which the coaching and scaffolding of apprenticeship-style learning are impossible (Collins et al, 1989). They argue that it is precisely in human-resource-intensive settings, such as tennis coaching, learning foreign languages at Berlitz, or receiving training in medical diagnosis, that apprenticeship methods are still used. They also propose that appropriately designed computer-based modelling, coaching, and fading systems can make a style of learning that was previously severely limited, cost effective and widely available. Again they recognise that, of course, apprenticeship-based computer systems need not take on the total responsibility. Instead, they only need to augment the existing training in a way that amplifies and makes it more cost effective.

7.2. How the identified knowledge representation paradigms and their associated knowledge types map onto the categories of knowledge recognised by the CAM

Each of our three types of knowledge identified in chapter 5, i.e. symbolic, experiential and metacognitive will be examined in turn to determine how they correspond to any of the types of knowledge considered within the CAM, described in section 7.1.

Our identified knowledge types:

(a) Symbolic Knowledge.

The category of domain knowledge, described in the CAM's content section which lists required knowledge, can be seen to map directly onto our category of symbolic knowledge, including both declarative forms such as the nature of physical structures and procedural forms such as low level design rules.

(b) Experiential

This type of knowledge is not explicitly referred to within the content section of the CAM. However, its existence does seem to be alluded to within other parts of the CAM. For instance, the least stressed of the two kinds of modelling deemed necessary by Collins (1991), when discussing the modelling and explaining section of the CAM, is the modelling of real-world processes, for example, the movement of electrons in circuits, Haertel (1987). The suggested need for the modelling of such knowledge seems to provide recognition of the need

for the type of knowledge we have categorised as experiential. Our interpretation also suggests that experiential knowledge is important for the enhancement of domain knowledge by means of intuitive learning.

(c) Metacognitive

The CAM's heuristic strategies and control strategies can be considered to come within our identified category of metacognitive knowledge.

The CAM divides the knowledge needed for expertise into explicit domain knowledge, which we have discussed in section (a) above, and also into various kinds of strategic knowledge. The term strategic knowledge is used in the CAM to identify the often tacit knowledge that provides the expert with the ability to organise the process of problem solving. This knowledge consists of problem-solving strategies and heuristics that can control the process of problem-solving through all levels of problem decomposition.

Our metacognitive knowledge includes such strategic knowledge and additional knowledge of wider activities that is needed to carry out civil engineering projects. Compared to the tasks considered in the development of the CAM, Civil Engineering projects are larger, demand longer time scales, require the generation of particular documents at specific times, need to be integrated with other peoples activities and also require approval at various stages by other personnel.

So, not only do we need to consider the short time scale (e.g. 1 day) problem solving strategies, for applying the domain knowledge, as considered in the CAM but also the higher level, longer time scale (e.g. 3 months) project structuring strategies that are needed in the Civil

Engineering domain.

Such "high level" metacognitive planning knowledge for the selection, structuring and integration of tasks to accomplish a large project is not evident in the previous applications of the CAM, probably because the activities considered by them are of a relatively short duration.

Thus our highest level of metacognitive knowledge concerns knowledge of the Civil Engineering project life-cycle, its phases, activities and deliverables. The project life-cycle is adapted according to the needs of the particular project being undertaken and specific elements of the life-cycle may also vary according to company practice. **Heuristic** strategies for adapting and applying the project life-cycle can be taught as rules. However, it is intended that the MLE will follow the principles of CAM to show the trainee the patterns of procedures and approaches which best suit particular real world problems. **Control strategies** are built into the overall structure of the project life-cycle and will be learnt with it. For instance, the **monitoring component**: production of deliverables, approval meetings; the **diagnostic component**: meetings and discussions to clarify difficulties; **remedial actions**: repeating a stage, abandoning a project. The project-life cycle used for the example project given within the MLE was developed from documentation of large scale projects undertaken by the "Master" Civil Engineer.

Our lower level of metacognitive knowledge involves knowledge of the strategies that are needed to accomplish an individual activity within the project life-cycle. An important example of such strategies is the sequence of application of the design rules that are needed to carry out the design process within a particular design activity.

The final element of metacognitive knowledge which needs to be represented is the contextual knowledge of civil engineering projects such as the documents required to record particular events and agreements and forms of communication between participants CAM implicitly addresses this need through the principle of Situated Learning.

The principle of situated learning is very well suited to the training practices for trainee engineers described in chapter 2. It suggests that an instructional strategy based on realistic case study problems, similar to those "real" problems given to trainees, would be successful. The social context of situated learning could be simulated by the MLE by means of videos. However, as engineers and students work in a real social context well supported by their training manager or tutor this is probably not necessary. Strategies for handling this aspect of training with the MLE are discussed in chapter 9.

Figure 17 illustrates the way in which the acquired knowledge and its representation styles map onto the CAM.

Thus the major knowledge content of the CAM approach is defined in two of its main elements, as shown in Figure 17, i.e. "Content" and "Modelling and Explaining" which correspond to conventional declarative text-book knowledge, various aspects of procedural knowledge including rules for design, modelling real world performance of physical phenomena, and metacognitive knowledge in the form of problem solving strategies. Knowledge of the content of projects is covered by "Situated Learning".

Together, these three aspects of the CAM are able to comprehend the types of knowledge representation required by the domain of dam spillway design.

Figure 17. How the knowledge representation paradigms of the domain of dam spillway design map onto the types of knowledge recognised by the CAM.

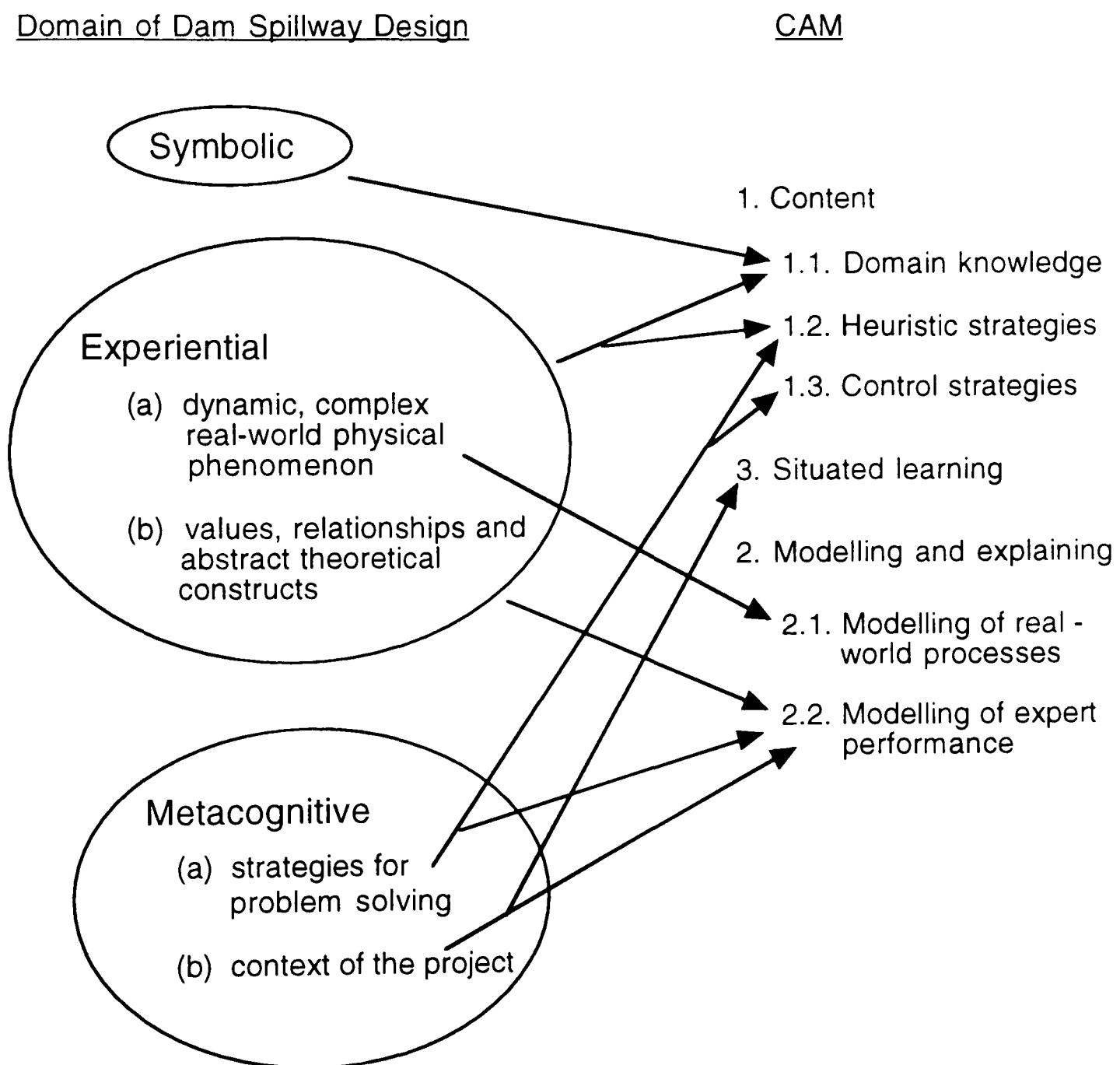


Figure 17 represents a simplified view of how the representation paradigms and some of the items of knowledge within them map onto the CAM. The Symbolic knowledge maps directly onto the CAM's Domain Knowledge, which is also supported by experiential knowledge to foster intuitive learning. The recognised need for Experiential knowledge of dynamic, complex real-world phenomenon corresponds to the CAM's proposal for modelling real-world processes. But this is also extended to the need to provide sensory experience of values,

relationships and abstract theoretical constructs by various multimedia means, such as, simulations, video, pictures, and so on. Most items of knowledge within the Metacognitive paradigm match the CAM's Heuristic strategies and Control strategies. Contextual knowledge is covered by situated learning. The general need to experience knowledge matches the CAM's modelling of expert performance. The teaching of the metacognitive knowledge and all our knowledge types is improved by situating the learning within a real-problem context.

The remaining elements of the CAM, as described earlier in this chapter, are as follows:

- 1.4 learning strategies
4. Coaching, including Scaffolding and Fading
5. Articulation
6. Reflection
7. Exploration
8. Sequence

All of these elements refer to the way in which knowledge is presented to the learner as part of a coherent pedagogic strategy, rather than the knowledge itself.

The following chapter illustrates the ways in which CAM's pedagogic principles have been interpreted for this particular domain by means of examples of elements of the MLE.

8. Design and implementation of the MLE

This chapter provides an outline design of the MLE which is illustrated with examples of implemented MLE components. Both the underlying architecture and the user interface are discussed. The chapter concludes with a number of brief comments on some of the problems encountered and some of the lessons learnt during the design and implementation of elements of the MLE.

From the discussion and mapping given in chapter 7 it can be seen that a passive base of domain reference material could be constructed from all the items of knowledge from within the symbolic and some items from within the experiential knowledge categories. Such material could be referred to and used to help solve a variety of problems within the domain. However, to experientially teach the domain's essential metacognitive knowledge and the CAM's category of learning strategies the MLE has to provide, in itself, an active strategy of learning through "real-world" problem solving that is based on the CAM and requires the learner to actively use the base of domain reference material.

The type of design activity which it is believed can be best supported by the MLE is the design activity needed for the solution of routine design problems. This is because novice engineers tend to be initially given routine design problems and routine design activity is better understood than creative design and must be initially mastered by the trainee graduate before they attempt moving on to more creative design. To limit the work to be done the "example design problem" used during knowledge elicitation was chosen for demonstration within the MLE. To promote learning of the symbolic and experiential knowledge by the MLE, elements of this knowledge would have to be referred to within, and needed for the solution of, the example design problem.

8.1. Illustrated schematic overview of the MLE

Figure 18. MLE Architecture.

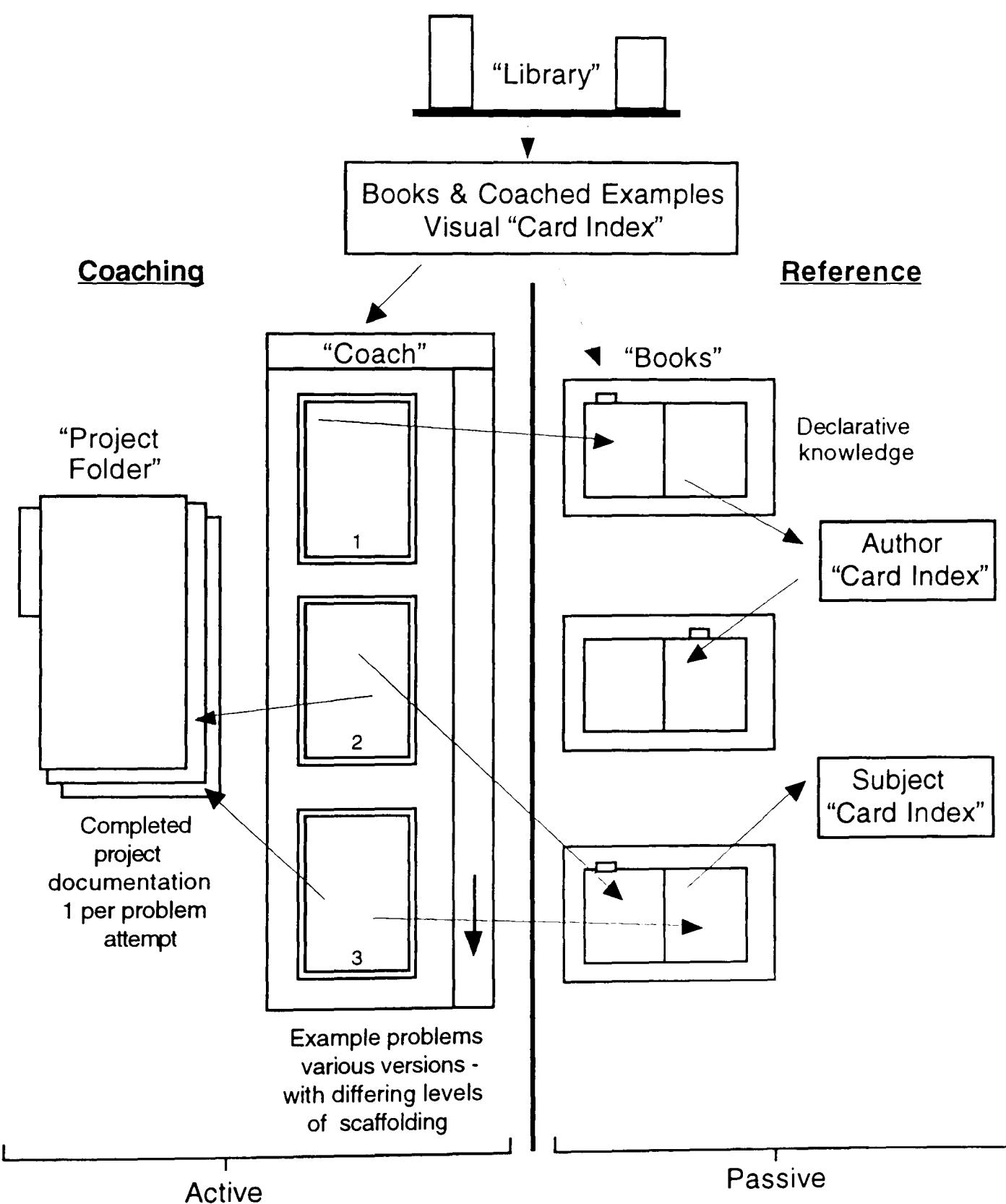
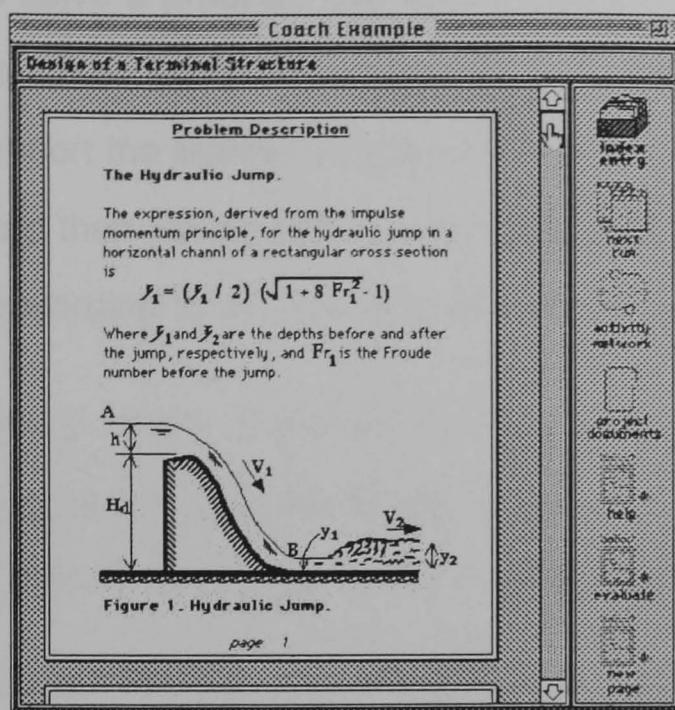


Figure 18 provides a schematic overview of the components of the MLE.

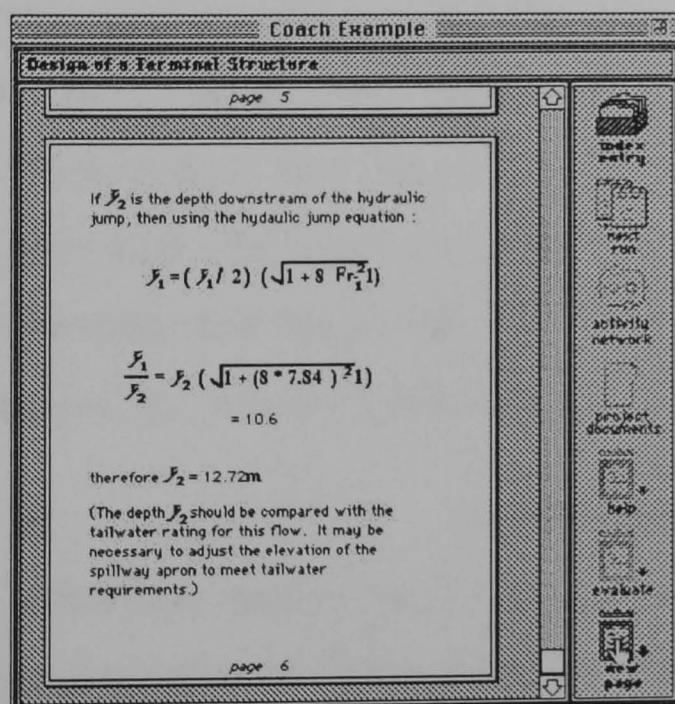
The "coach" first models the expert solution of a problem. The learner is shown the problem solution, sequentially, a page at a time (for example, Figures 19(b) - 19(c)). Pages already seen can be re-examined by being scrolled through in the

Figure 19. Example screens from the "Coach".

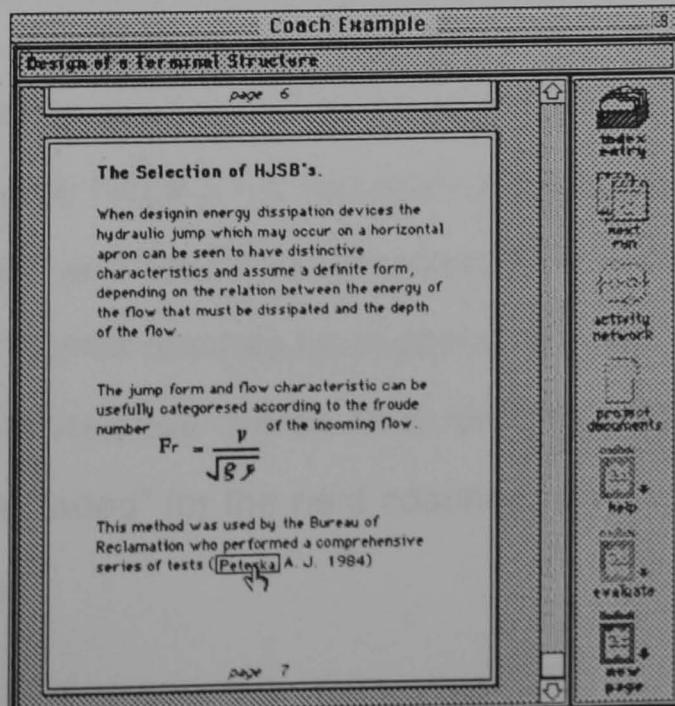
19(a)



19(b)



19(c)



same way as a word processor (for example, Figures 19(b) - 19(a)). After being shown how to solve a problem, the learner can then sequentially work his or her way through a similar type of problem but one which uses different values. The system will support the learner by alternately demonstrating the solution to part of the problem and then asking him or her to attempt part of the problem, and so on. This is done according to the “global before local” sequencing of skills.

As a case study problem is worked through, values are filled in on examples of project documentation held within the “project folder”. This provides students with a correct model of the appropriate ways of recording their findings.

“Books” of reference material are referred to in the coach (for example, Figure 19(c)) to provide background understanding for learners and to help them complete their parts of the task. The “books” are structured and contain organised multimedia explanations and demonstrations of symbolic and related experiential knowledge which can be used to solve many different types of problem (for examples, see Figure 20). The “books” contain hyperlinks and so allow the learners to “explore” the material and find out information for themselves.

“Books” are referenced within “card indexes” and the coached examples themselves are also initially taken from an “index”. All “books”, “coached examples” and “indexes” are available from within a “library” (Figure 21).

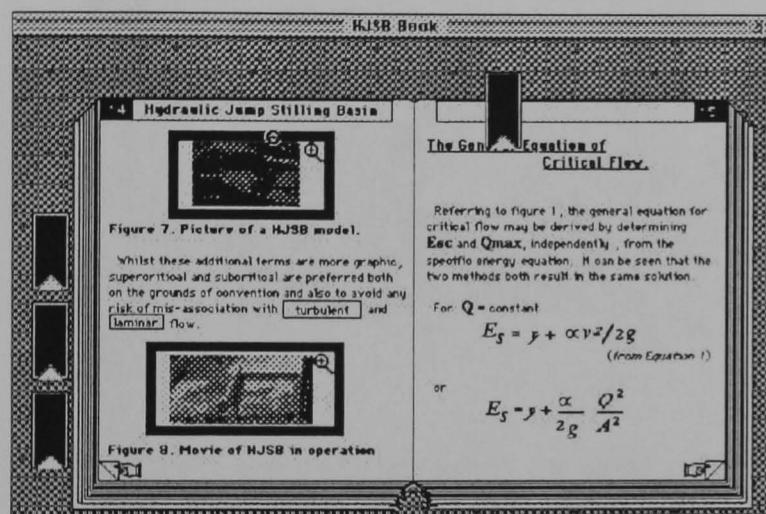
After completing a section, learners ask for it to be evaluated. If they have done badly, they can be helped. For example, the system can demonstrate the solution to that section, and levels of demonstration with varying detail can also be shown. If the learner reaches an impasse, they can, again, be given similar help. How well a learner does overall on a run-through determines how much of the “scaffolding” is “faded” for the next coached run-through of a similar problem with different values.

Figure 20. Example screens from a “Book”.

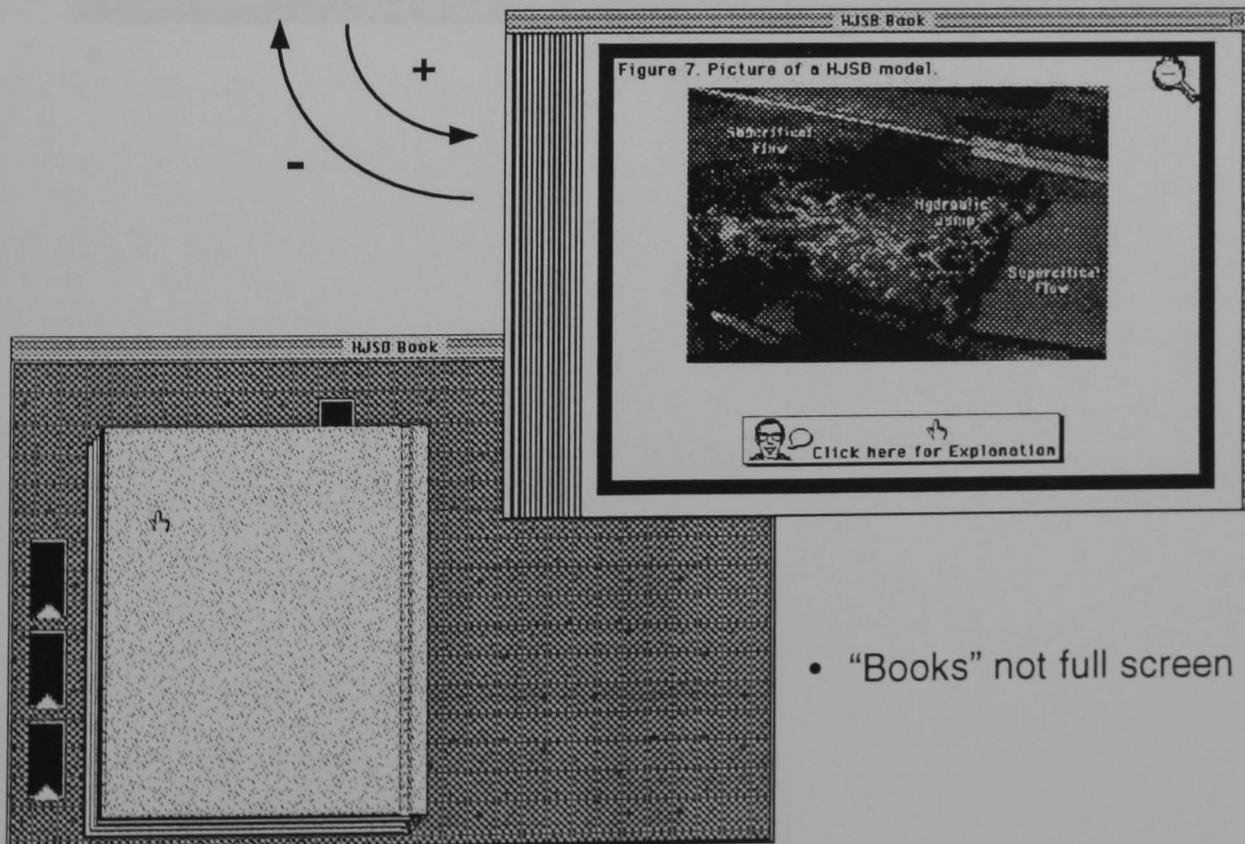


- Look and feel of a real book.

User can: turn book over, flip pages, turn a page and drag bookmarks.



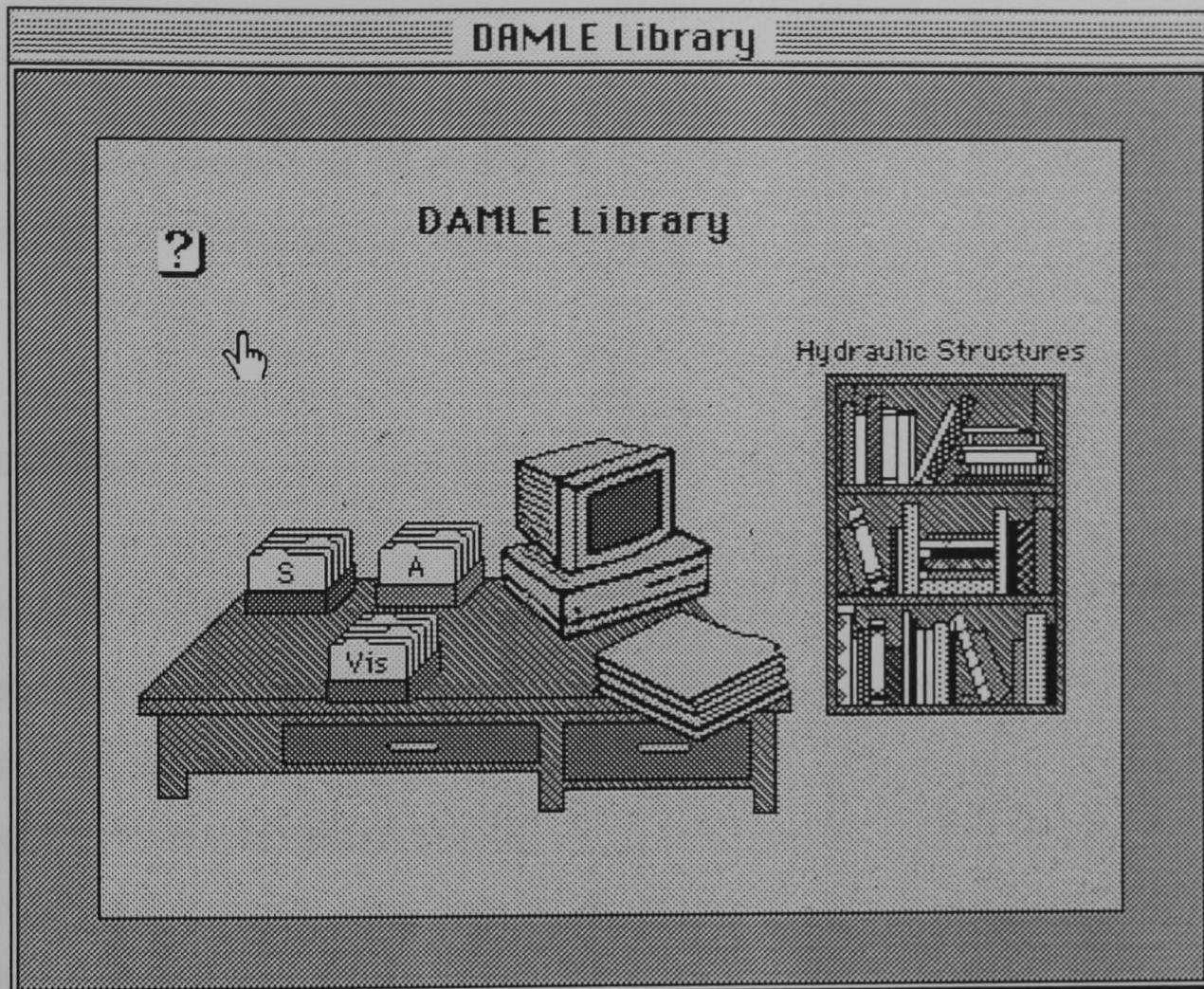
- Some items, e.g. pictures, video and simulations, that are too big to be placed on the page full size, appear shrunken on the book and can be zoomed in and out upon.



- “Books” not full screen size.

- The “coach” and a number of “books” can appear on screen at once (overlapping).

Figure 21. Example screen from the "Library".



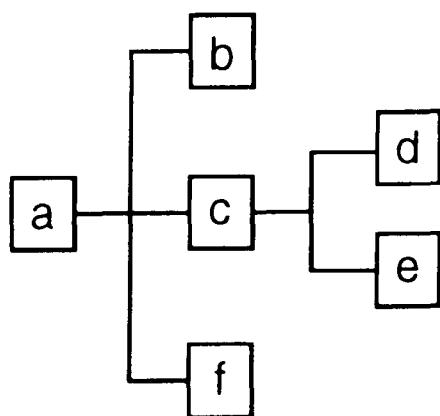
8.2. Underlying design of the MLE: mechanisms needed to realise the major elements of the CAM

The following is a simplified outline of the underlying architecture and mechanisms needed to implement the CAM's core elements of modelling, coaching, scaffolding and fading.

Modelling

Each of the tasks in Figure 22, which shows the decomposition of a simple problem into individual steps, would be represented as a node within the system. Each node has a set of “system demonstration pages” and a set of “learner work-through” pages associated with it. The system demonstration pages illustrate how an expert would solve the task (it includes expert “articulations” and “reflections” on their problem solving), and the learner work-through pages invite learners to solve the task for themselves. Initially, the expert’s solution to the problem, using a particular set of values, is modelled. This is accomplished by using a stack to deliver the system demonstration pages for each node, in turn, i.e. a, b, c, d, e and f.

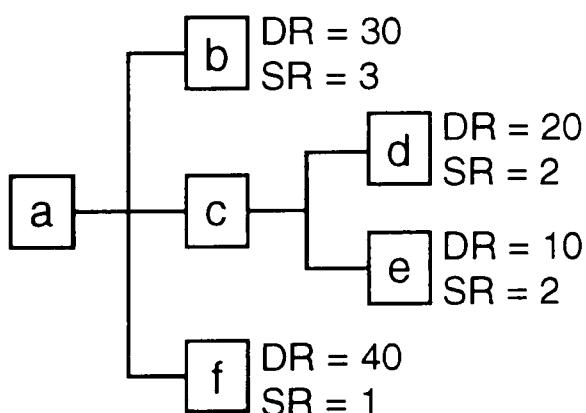
Figure 22. A generic design problem decomposed into a number of tasks.



Scaffolding and fading

Each leaf node in Figure 22 (b, d, e and f) also has an evaluation procedure, a learner score and a difficulty rating and sequence ranking (see Figure 23, also, see Figure 24 for the decomposition of the tasks required to perform the example design problem). A task's difficulty rating is a measure of how complex and time-consuming a task is to complete. The task sequence rankings are used in the process of allocating the next task to be attempted by the learner.

Figure 23. Design problem nodes with example difficulty ratings and sequence rankings.



DR - Difficulty Rating, SR - Sequence Ranking

(A “real” problem would contain many more tasks, for example, see Figure 24).

On a “scaffolded run-through”, the same design problem is used but with different values, so different choices may have to be made. For example, in the example civil engineering design problem, different types of spillway stilling basin may need to be considered.

When “scaffolding”, the system will demonstrate some parts of the problem, and the learner will attempt other parts of the problem. The basic method for allocating tasks to the learner uses the learners scores, difficulty ratings and sequence rankings, as outlined overleaf:

- Leaf nodes are ordered according to their sequence rankings and difficulty ratings: f (SR = 1, DR = 40), e (SR = 2, DR = 10), d (SR = 2, DR = 20), b (SR = 3, DR = 30).
- Say 50% of the given problem (in terms of difficulty) is to be allocated to the learner on the next run-through (the higher a learner's overall score on a run-through, the more of the problem they will have to attempt on the next run-through).
- 50% of the total difficulty of the whole problem is calculated, i.e. 50 (in this case). So, a number of tasks with a difficulty totalling 50 (or as close to it as possible without exceeding it) will have their node settings set to show learner work-through pages. Thus, during the run-through, at these points, the learner will be asked to attempt those tasks on his or her own.
- The first tasks whose difficulty ratings total 50, or as near as possible to 50, are allocated to the learner according to their sequence rankings. So, on the next run-through, nodes f (SR = 1, DR = 40) and e (SR = 2, DR = 10) will be allocated to the learner to attempt.
- The sequence rankings are intended to deliver the CAM's "global before local" sequencing because the centrally important, guiding tasks and decisions will be given higher sequence rankings in order to ensure that learners will practice them earlier.

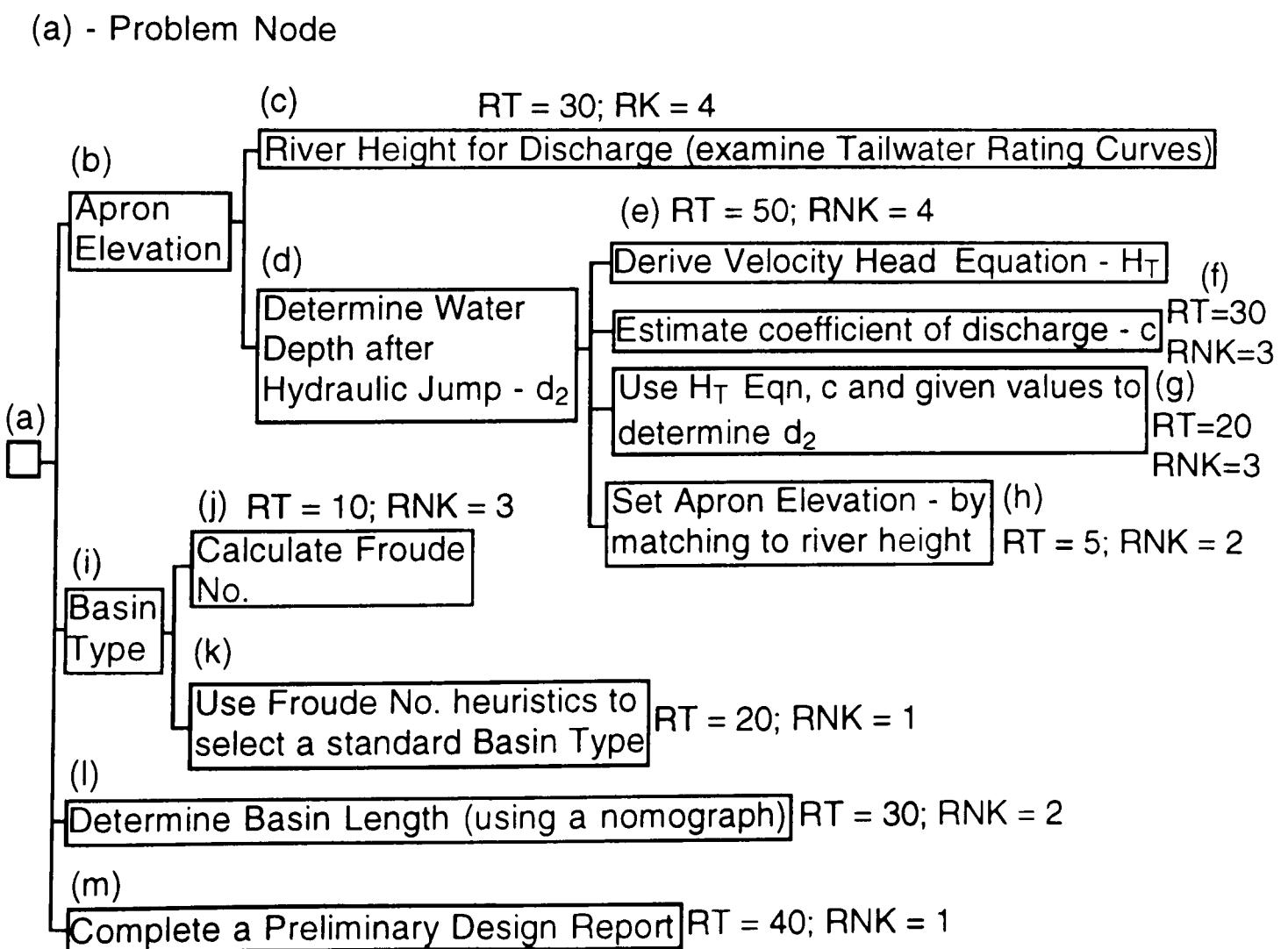
Coaching

Each section attempted by a learner is evaluated (by considering the values calculated, the design choices made and by using multi-choice questions to

probe the learner's articulation and reflection on a problem). If the learner does badly on a section, he or she can be shown the system demonstration pages for the node concerned, its child nodes and so on (varying demonstration/help pages can be given depending on the errors made). If a student is unable to continue on a section, again he or she can be shown the system demonstration pages for the node concerned and its child nodes (and, again, varying demonstration/hint pages can be given depending on the experience of the learner, for example how much of the problem he or she has been allocated).

Decomposition of the tasks required to perform the example design problem

Figure 24. Decomposition of the tasks required to produce a preliminary design for a hydraulic jump stilling basin.



RT = Difficulty Rating; RNK = Sequence Ranking

8.3. How the human tutor may use the MLE: inherent limitations of the MLE

It is envisaged that a situation may arise where the human tutor needs to teach the MLE's subject matter to the target learners. It is anticipated that the tutor will already have become familiar with and have realised the value and potential of the CAM. In such circumstances the tutor will view the MLE as being an additional resource possessing, like any teaching aid, its own inherent limitations.

Comparing the CAM based design of the MLE specified in the previous section to the comprehensive description of the CAM detailed in chapter 7 reveals a number of areas where the MLE alone will be unable to do full justice to the CAM, in particular:

- (a) Coaching will suffer as the MLE will only be able to offer an extremely restricted set of pre-determined, fixed responses to the learner's sub-problem solutions or requests for help. In addition, it will not be possible to receive even this limited coaching feedback while actually trying to solve a sub-problem as the MLE only asks diagnostic multi-choice questions once after an attempt at a sub-problem has been completed.
- (b) The multi-choice questions will be used to try to determine the validity, and the learner's understanding of the problem solving control strategies used, the final values calculated and the design decisions made. However, the MLE can make no direct assessment of the learner's supporting calculations as they will only be written down paper. Also, the MLE will make no attempt to analyse the findings and conclusions which the learner will type into the computer-based

project folder.

- (c) Initially, only one relatively straightforward routine design problem will be available within the MLE.
- (d) The learner will only ever receive one, i.e. the MLE's, point of view.

For the above reasons, to successfully convey the subject material using the CAM, a human tutor will also be needed. The tutor will be able provide a more accurate diagnosis of the problems which may arise as the learner is working through their solution. The tutor's help and feedback will also be far more flexible and appropriate. In addition, the tutor will be able to develop and teach a suitable sequence of increasingly complex and diverse design problems. It will also be useful to involve other tutors and fellow learners to give alternative perspectives on solving the same design problems.

So, an integrated approach will have to be used. For example, the tutor may begin by modelling the solution to a design problem to an individual learner or group of learners either on paper or by using a white board. The tutor would use the CAM approach to modelling, including verbalising their thought processes as they progressed, formulating and trying out hypothesis, and backtracking if necessary. The tutor would also complete a final design report summarising findings and recommendations. Finally, the tutor would conduct a reflective analysis of the complete solution by retracing the solution method and recalling any heuristics used, points where alternatives were generated, why a particular alternative was selected, and so on. Then one or more learners will each have to attempt to solve parts of similar problems in the same fashion. The tutor or indeed other tutors or fellow students, if present, would be able to offer hints, reminders and feedback as needed and they would also be able to provide a critique of the completed solution and solution method.

Following this introduction to solving suitable types of design problem and the CAM the learners will then be required to use the MLE on their own to continue practicing and learning the relevant domain knowledge and skills. More CAM based tutorial sessions may follow with the tutor using increasingly complex, diverse and creative design problems. The ultimate goal will be the learner submitting, for assessment, completed design problems which include all workings, explanations, reflections and concluding reports. Although the MLE cannot fully replace human tutors and fellow learners, when it is used it should still be a useful personal guide and be able to provide some valuable practice and background knowledge. The learner will soon realise the MLE is unable to directly check supporting calculations, articulations, reflections and design reports but it is hoped the learner will nevertheless attempt to do these as fully and competently as possible as they will appreciate that they will need to be able to do these in tutorials and for assessed work.

An additional sub-program will be needed to enable the human tutor to generate adjusted difficulty ratings and sequence rankings for the nodes of the MLE's example problem. These values will be adjusted by the tutor if they feel the learners need more practice at solving certain sub-problems. To ensure the MLE does not behave unexpectedly only values within pre-determined ranges will be allowed and guidance about the effects of altering the values will be given.

It would also be useful if this sub-program could be used by the tutor to allocate and record the work undertaken by each learner. Initially only differently adjusted versions of the one in-built example problem could be allocated but it is hoped that more problems will eventually be added to the MLE. Learners could be listed and allocated to groups and the groups or individual learners allocated problems. If the problems had been adjusted by the tutor the files generated by the sub-program which would contain the new difficulty rating and sequence ranking node values will have to be made available to the appropriate learners

as they will be using an original copy of the MLE from their own CD-ROM and this copy will need to read in these values to make the correct adjustments. A learner will not have successfully completed an allocated problem until they have worked through a number of scaffolded, slightly different run-throughs of the problem and been able to correctly complete a whole problem without assistance from the MLE. To produce each slightly different run-through of the same problem the MLE will randomly generate, within specified ranges, the initial values for each run-through and then insert these values into the problem's pre-determined, fixed problem template and design rules. Once a learner has successfully completed a problem the tutor should be able to record this within the supplied additional sub-program. If the tutor wanted to ensure the work had really been done the tutor could check disk copies of the learner's completed "Coaches" and "Project Folders" which would be generated by the MLE for each run-through. If a large group of learners were involved this could be done at random.

The above is just one example scenario of how the MLE may be treated as another resource by the human tutor and integrated into existing teaching while taking into account the MLE's inherent limitations.

8.4. Authoring systems used and technical problems encountered

When it became necessary to begin developing the prototype the most appropriate hardware and software to use had to be chosen. An Apple Macintosh was used because during this period, towards the end of 1992, the Mac was widely acknowledged to be the premiere multimedia development platform. At the time this often resulted in multimedia products that were only ever destined to be released on the far more widely used IBM compatible PC nevertheless actually being developed on a Mac

Several authoring tools for the Mac were considered including: *Authorware 2.0* which used a “flow-line” metaphor; *Director 3.0* where elements were placed on a “time-line” and *SuperCard 1.6* and *HyperCard 2.1* where in both, applications were constructed from “stacks of cards”. It was realised that any authoring tool chosen would have to be suited to and capable of implementing the design envisioned in the earlier sections of this chapter. In addition, due to the limited budget for this research, the authoring tool chosen could not be too costly and would have to be able to run effectively on a relatively low-end machine.

HyperCard 2.1 was chosen because its card and stack based metaphor and its underlying “BASIC-like” scripting language was thought by the author to be well suited to the envisaged design which required pages of mostly static material, including some user initiated multimedia elements, which had to be linked together and moved between in various ways. On a base level machine it was also relatively fast and stable and required a comparatively small amount of RAM and hard disk space. It was also inexpensive and as it had been given away free with each Mac from 1987 to the beginning of 1992 there was a large installed user base with many relevant books, example programs and third-party enhancing, add-on external commands and functions all being readily available.

So, prototype implementation began using *HyperCard* 2.1. Throughout prototype development, additional software and hardware was also used to create and prepare individual multimedia elements, for example, *Premiere* (video digitizing and editing), *Photoshop* (scanning, manipulating and creating images), *Director* (creating simple animations), *SoundEdit* (creating equations), video cameras, video recorders, still digital cameras, scanners and microphones. Much of this hardware and software was available within refurbished labs commissioned for the University of Plymouth's new MediaLab Arts degree course.

In an attempt to do justice to the metaphors conceived in the design a great deal of time was spent trying to improve on some of *HyperCard* 2.1's in-built interface and multimedia facilities, for example, for hyper-link choices fast acting, intuitive and consistent menu selections which popped-up on text were painstakingly created, great pains were also taken to ensure pictures and video clips which had to be shown in a separate, independent, floating window actually appeared as if they were within and moved with their "book" or "coach" page. However, much of this effort was wasted as the later edition of *HyperCard*, *HyperCard* 2.2, released a year or so later was a major revision and included facilities to accomplish these tasks, for instance, it had built-in pop-up menus and came with external commands which enabled pictures to be copied or loaded directly onto a card. Although this later version brought with it incompatibilities with the earlier edition, for example, the extremely effective built-in page turning visual effect exploited in the "books" could no longer be used once the bundled external command had been invoked to incorporate colour pictures.

A radically updated version of *SuperCard*, *SuperCard* 2.0, was released a year or so after *HyperCard* 2.2's release. *SuperCard* had originally been derived from and in some areas had been an improvement upon *HyperCard*. *SuperCard* 1.6 had not initially been chosen because it was extremely slow and its future seemed uncertain. However, its original developers had since left Aldus to form

their own company, Allegiant, to concentrate on improving and marketing *SuperCard*. The result was *SuperCard 2.0* which was PowerMac native enabling it take full advantage of the new faster Macs. As a consequence, on such machines, it was considerably faster than *SuperCard 1.6*. Numerous well thought out improvements had also been closely integrated into *SuperCard* itself, for example, the page turning visual effect could still be used when colour pictures had been incorporated. Allegiant also announced that *SuperCard 2.0* applications would soon be able to run from within a web browser and that they were working on a fully compatible PC version of *SuperCard*.

In contrast, the improvements to *HyperCard*, from 2.1 to 2.2 and, by this time, the recent minor improvements arising from release 2.3, were largely the result of including separate, clumsy and awkward to use external commands with the software. In addition, Apple appeared to have no immediate plans to make *HyperCard* PowerMac native (although they soon intended to replace the whole of their Mac range with PowerMacs), more closely integrate existing improvements, allow *HyperCard* to run within a web browser or develop a fully compatible version of *HyperCard* for the PC.

As a result, for these reasons and the fact that *SuperCard 2.0* was structurally similar to and still retained many of the same scripting constructs as *HyperCard 2.3*, it was decided to use *SuperCard 2.0* to implement components of the MLE which had not yet been fully implemented in *HyperCard 2.3*. The MLE's "Books" and "Coach" remained in *HyperCard 2.3* while the "Library", "Card Indices" and "Project Folders" were implemented in *SuperCard 2.0*. When the MLE was executed both *HyperCard* and *SuperCard* would automatically open and communicate with each other via Apple Events to open windows, turn pages and so on. The layout of both program's limited menu bar was the same and consistent colours and styles of components had been used within each window. The aim was to give the appearance to the learner that they were only using one

application. Eventually, it was thought it might be advantageous to implement the whole MLE in *SuperCard*.

There were some problems common to both *HyperCard* and *SuperCard*. Both tools had limited text handling capabilities which meant each equation had to be created in another package, Word, and then pasted into the MLE as a graphic. This caused problems as it precluded appropriate textual explanations containing equations being read in on demand from external files and positioned and displayed when needed. Instead, pages containing every possible combination of pasted together explanations had to be pre-produced to be shown if needed. In general, it was cumbersome and constraining to have to convert media elements created in other packages into formats acceptable to *HyperCard* or *SuperCard* and then have to paste those elements directly into the *HyperCard* or *SuperCard* document. Once a media element was pasted in it would be difficult for the developer of the MLE, or any other developer, to retrieve the media element in order to edit or copy it to use it elsewhere. It was also extremely difficult to rapidly perform complex calculations and rule checking with the simple scripting languages underlying both packages. From initial investigations, converting code produced in Pascal or C into appropriately formatted, compiled and usable external commands and functions appeared excessively complicated and was not attempted.

8.5. Formative Evaluation

To improve the prototype, during each stage of its development various evaluation techniques were continually applied. It was deemed necessary to attempt to evaluate the three important aspects of the MLE considered below:

- (1) the validity of the content of the MLE
- (2) the usability of the interface components of the MLE
- (3) the teaching effectiveness of the MLE

Initially, the validity of the proposed content of the MLE was confirmed as the knowledge was being acquired by using the techniques discussed in 'Chapter 4. Knowledge acquisition'. A restricted form of Protocol Analysis and the Teachback Interview, both detailed in section 4.2.1., were employed at an early stage to ensure, from the Journeyman's point of view, that the developer had correctly understood, applied and specified the appropriate domain knowledge. This knowledge, described in sections 4.2.2. and 4.2.3., was verified and added to, as explained in section 4.3.2., by informal discussions with the Master, as outlined in section 4.3.1. Supplementary information was also elicited and authenticated from interviews with industrial practitioners, as examined in section 4.4.

Throughout the implementation of the prototype many informal, group usability inspections were carried out in a style which, in hindsight, can best be described as a simplified form of Pluralistic Walkthrough (Mack and Nielsen, 1994). Each walkthrough involved a meeting between the developer and some or all of the members of the developer's supervisory team. At each meeting the developer would begin by outlining the aim of the current evaluation and describing a realistic user scenario that would be considered for the purposes of the

evaluation. The developer would then present what would initially be seen and heard on screen, any parts not yet fully implemented would be substantiated by showing detailed paper designs and the developer would also be able to provide verbal clarifications. The other participants would each assume the role and objectives of the user portrayed in the user scenario. In their role as the user each participant would, in turn, describe how, in order to achieve the user's objectives, they would interact with the screen presented and what they would then subsequently expect to be presented with, for example, anticipated information, media items, encompassing metaphors. During this the developer would take notes and once all the participants had finished the developer would put forward what the developer thought the user would do and then present what would be the resulting subsequent screen. Any discrepancies between the participants and the developer would form the basis for a discussion on possible re-design. This would continue until every screen involved in the proposed scenario had been presented and discussed in sequence.

By performing the walkthrough potential problems overlooked by the developer were uncovered and the developer was able to gain new perspectives on and understanding of the proposed design and implementation. In addition, during the discussions of the problem areas the other participants were often able to suggest possible improvements to the developer and, in turn, the developer was often able propose alternatives and then be provided with immediate feedback from the other participants. The contrasting expertise of each participant provided valuable alternative perspectives on each problem area, for instance, the Journeyman was most concerned with the accuracy and relevance of the content and the general suitability of the MLE for use in an engineering practice, the Reader in HCI with the usability of the interface and the MediaLab Arts degree course-co-ordinator with the quality and effectiveness of the media elements. As all the supervisory team were also experienced lecturers they were all concerned with the effectiveness of the teaching strategy and its appropriateness

for undergraduate training. The walkthrough method would have been even more valuable if typical end users had been involved. However, this proved difficult to arrange.

As part of another project (Parsons, 1994) a limited user evaluation of one of the MLE's "Books" was carried out. The evaluation was chiefly intended to reveal usability problems with the implementation of the "Book" metaphor, although some feedback on the relevance and usefulness of the information was received from the Civil Engineering evaluators involved, and comments on the quality and effectiveness of the media elements were received from these and the other evaluators. Six students were used as evaluators, three undergraduate Civil Engineers (all final year students) and three students from different disciplines. They ranged in computer experience from novice to proficient. Only one of the evaluators had previous experience of using an Apple Macintosh. The evaluation lasted for approximately one hour for each subject. The subjects were first introduced to the purpose of the evaluation and its structure. They were then given a personal questionnaire to complete which asked about their previous computing experience and whether they had previously used any computer based training packages. They were also asked to rate their existing knowledge on three topic areas which were covered within the "Book". They were then given a specific question about the domain and asked to browse through the "Book" at their leisure to try to answer the question. After browsing, each student was given a structured interview where they were asked to provide their answer to the initial question and to again rate their knowledge of the three topic areas. They were also asked a number of very specific usability questions about areas the developer thought might be problematic, and asked to give some general feelings about broad issues such as how interesting, involving, useful and consistent they found the "Book" and how "in control" they felt while using it. As well as this they were asked how suitable for the subject matter they thought the metaphor was and to rate the quality of the videos, animations, and explanatory

narrations. The Civil Engineering students were also asked if they thought the information they found was correct and whether they would rely on it and use it in the workplace. All the students were asked for any other comments which they thought would be of interest to the developers. The evaluation highlighted in detail a number of usability problems which were then worked on, for example, the quality of the movies were improved, the video narrations were extended, in places the text was enlarged, more detail was put into the table of contents and the pages concerned with the derivation of equations were removed to the back of the “Book” to be referenced only when needed. More generally, the evaluation revealed that the subjects believed the “Book” to be a pleasing, inviting, easy to understand, consistent and useful metaphor within which they felt completely in control. The Civil Engineering students believed all the content material to be relevant and, as far as they could tell, correct. The students were all particularly impressed by and valued the use of video and especially the use of interactive animated diagrams. All the students thought they would use such a “Book” in the work place. Most of the students also answered the initial question correctly and they all felt more confident about the subject matter after using the “Book”. For a more detailed analysis and discussion with the original answers to each student’s structured interview questions see Parsons 1994.

Once a realistically usable proportion of the coach and other interface elements had been implemented it was intended to carry out more extensive evaluations with typical users.

9. Conclusions

9.1. Overview

The aim of the research was to develop a better understanding of the ways in which interactive multimedia learning environments (MLEs) can aid learning, with particular reference to the qualities of multimedia which make MLEs different from CAL. In practical terms the intention has been to make a contribution to the methodology of MLE design. At the most fundamental level this has included a reinterpretation of some of the basic principles of human learning, knowledge representation and knowledge engineering in order to characterise in psychological terms some of the new affordances which are offered by interactive multimedia. This reexamination of the theoretical bases of learning has been necessary in order to provide as good a foundation as the imperfect science of psychology allows for recommendations for design method. This aspect of the work has reemphasised the complexity and variety of human learning and shown the clear need for a rich mixture of theoretical perspectives in presenting learning material, even in a relatively straightforward and well-defined domain.

Secondly, in recognising the essential role of cognitive science, the work has drawn upon techniques from knowledge engineering such as knowledge elicitation and knowledge analysis in order to ensure that the richness of the domain can be captured from experts and expressed to the learner through interactive multimedia.

Thirdly, the work has abstracted some of the generic ideas which have arisen during the project in order to make the findings more widely useful.

9.2. Methodological perspective

The contribution of the research can be seen in more detail from a perspective which follows the longitudinal progress of the work as a whole, and within it the steps of the MLE design process.

Chapter 1 overviews the project as a whole, and chapter 2 provides a description of the domain and its learners which were used as a vehicle for the work.

Chapter 3 reviews a representative range of computer-based learning products in order to substantiate the premise of the research that the methods, techniques and models used by other forms of CBL are not adequate for the design of MLEs. Section 3.4. summarises the main elements which were seen to be required for the MLE design process, and thus sets the agenda for the remainder of the work. The practical steps of a proposed method are outlined in Chapter 1, page 8.

Chapters 4, 5, 6 and 7 describe the steps of the design process which draw most directly on techniques of psychology and knowledge engineering including knowledge acquisition, knowledge analysis, pedagogic models and instructional strategies. These chapters show most clearly the application of different levels of analysis to each step of the design process, namely:

- reinterpretation of fundamental principles from cognitive science (including psychology, AI, software engineering and engineering design);
- selection of existing techniques and methods based on those principles;
- adaptation and articulation of principles, techniques and methods in terms of the specific practical needs of this multimedia learning environment;
- abstraction of principles, techniques and methods in terms of the generic problems of MLE design.

It is this multi-level analysis of the design process which has provided most of the

fresh insights which have emerged. Some of the more exciting examples have been:

Chapter 4

- The value of different expert perspectives on the problem domain and the characterisation of the different types of knowledge which they bring.
- The importance of experiential knowledge in “real world” learning, particularly in view of the ability of multimedia to simulate some kind of experience.
- The role of metacognitive knowledge in enabling the engineer to organise and apply knowledge effectively.
- The value of models of the engineering design process as descriptions of how engineers solve problems.

Chapter 5

- The analysis of domain knowledge in terms of Symbolic, Connectionist and Metacognitive paradigms of human knowledge representation. This analysis is particularly useful in providing MLE designers with a clear rationale for the way in which knowledge is represented to learners.

Chapter 6

- The convergence of different models of instructional design and their relationship to models of human knowledge representation.

Chapter 7

- The interpretation of the Cognitive Apprenticeship Model in terms of the types of knowledge identified in the learning domain and models of human knowledge representation.

Chapter 8 describes the final design of a prototype MLE which has benefitted from this approach.

Needless to say, this thesis represents a highly sanitised account of what was in practice a very messy and convoluted exercise.

9.3. The Knowledge Engineering approach

In evaluating this approach we are implicitly comparing it with conventional methods for the design of computer based learning which in general are less inquiring and tend to focus more on the production process, for example those described in section 3.3.

The essence of knowledge engineering, as we have interpreted it, is to consciously seek alternative explanations and representations of knowledge based on evidence from psychology and cognitive science. This quality has provided a number of benefits.

First, it has led us to recognise that in this domain and probably others, different categories of expert exist, each with their own perspective and priorities. Including these multiple views can enrich learning in important ways.

Secondly, analysis of the domain knowledge in terms of its underlying representational paradigms has alerted us to qualities and categories of knowledge that were not evident at face value. Greater consciousness of the knowledge has enabled the designers to capture and express aspects of the domain which would probably have been underplayed otherwise, for example the experiential and meta-cognitive elements.

Recognition of experiential knowledge has particular significance for multimedia because of the pattern-matching nature of such knowledge and the great power of multimedia to display visual images, both moving and still, to communicate informal intuitive understanding and conceptual prototypes. This finding in particular has confirmed our original hypothesis that multimedia can support some types of learning much more effectively than the older technologies of computer based learning.

Thirdly, a knowledge engineering approach has provided the designers with a touchstone to which they could return when seeking clarification on the nature of the domain knowledge and how it should be represented. This arose from the implicit principle of knowledge engineering that design should be based as far as possible on psychological evidence, or at least widely accepted hypotheses, about human knowledge processing. Such evidence helped to clarify objectives and provided a rational basis for making choices and decisions.

In other words, a knowledge engineering approach encourages reference to a theoretical basis, however imperfect, for design. The process can be summarised as follows:

- (1) identify the learning effects which need to be achieved
- (2) identify the models of learning which describe this effect
- (3) identify the multimedia techniques which support this model

9.4. The Cognitive Apprenticeship Model

Several features of the CAM have made it an ideal instructional model for the design of this particular MLE and undoubtedly many others. The knowledge types it uses enabled direct mapping from much of the domain knowledge. Further, the apprenticeship metaphor was found to be particularly appropriate for the target learners.

As discussed in detail in chapter 8, ‘Section 8.3. How the human tutor may use the MLE: inherent limitations of the MLE’ the CAM as implemented within the MLE had many limitations. To do justice to the CAM and successfully convey the subject matter to the learner using the CAM it was realised early on in the design that the MLE would have to be just one part of a teaching strategy involving a human tutor and perhaps other human tutors and fellow learners. An example of one such integrated approach to teaching the domain material to the target learners was also discussed in detail in section 8.3.

To a certain extent the implementation of the CAM within the MLE may be improved by adding more “intelligence” to the MLE. For example, intelligent tutoring system techniques could be used to produce a component capable of generating and more flexibly and realistically solving a variety of increasingly complicated routine design problems. Another component could be developed to more effectively question the learner about their solution, to analyse and compare their answers more closely to the problem solving component’s solution to provide better information for determining how to continue with the CAM’s elements of coaching, scaffolding, fading and so on. However, there are numerous problems highlighted in chapter 3, ‘Section 3.1.3. Intelligent tutoring systems’ which, for this domain at least and in the foreseeable future, make the idea of attempting to take the implementation of the CAM further in an effort to completely replace the human tutor an extremely difficult, time consuming and

misguided endeavour. For instance, tutors and fellow learners can be very good at, and can themselves learn a great deal from, the process of diagnosing and helping an individual to overcome their shortcomings. Clearly, in this context, some teaching tasks can more appropriately be allocated to the tutor, fellow or individual learner, than to the MLE.

9.5. Possible directions for improving the design and implementation of the MLE using newly available technology

As discussed in Chapter 8, 'Section 8.3. How the human tutor may use the MLE: inherent limitations of the MLE' one major possible improvement would be to operate the MLE over a network. If this was accomplished the human tutor would find it far easier to check their learners had completed the work they had said they had completed using the MLE and the learners would find it easier to collect the tailored MLE problems they had been allocated. If maintaining a resource base of reusable individual multimedia items in their original format (a requirement mentioned in Chapter 8, 'Section 8.4. Authoring systems used and technical problems encountered') was an important consideration then an open hypermedia system such as Microcosm (discussed in Chapter 3, 'Section 3.1.4. Hypertext/Hypermedia') may be the best choice. However, if widespread distribution of the material was paramount then an obvious choice would be the World Wide Web. Over recent years the increasing use of the Web and the availability of suitable enabling technology has led many developers to transfer their existing standalone CAL packages onto the Web, for example, courseware developed for the Glasgow University Teaching with Independent Learning Technologies project (Creanor, Durndell and Primrose, 1996) has been transformed from Guide through Toolbook to the Web.

If the Web was to be used then the information within the supporting background material component of the MLE, that is, the "Books", "Card Indices" and the "Library" could be taken out of those metaphors and placed directly onto Web pages accessible to the appropriate learners from the human tutor's home page. Alternatively, the information could be kept within applications which maintained the metaphors and these entire applications then embedded into web pages, this would probably be the best option for the "Coaching" and "Project Folder"

components. Any material that was too slow to load and run over the Web, for example, large video clips could be stored locally on CD-ROMs.

SuperCard could be used as the embedded application because *SuperCard* programs can now be played from within a Web browser, *Netscape*, on either a Mac or PC providing the browser was equipped with a specially developed plug-in, *Roadster*. So, all the components would need to be converted into *SuperCard*. However, the recent very poor financial performance of the makers, Allegiant, resulting from uncertainty over the future of Apple and also Allegiant's inability to deliver on their promise to produce a version of *SuperCard* for the PC would mitigate against this.

In addition, if placing the MLE on the Web were to be taken as an opportunity to improve upon the design and implementation, for instance, to enable human tutors to easily add their own problems and background material (a requirement cited in Chapter 8, 'How the human tutor may use the MLE: inherent limitations of the MLE') then it may be wiser to choose a more powerful application with a more secure future such as one of the modern, widely used, visual programming tools, for example, *Visual Basic*, *Visual C++*, *Visual J++* (all from Microsoft) and *Delphi* (from Borland).

Improved multimedia and Web facilities are increasingly being incorporated into these tools. For example, *Visual Basic 5* applications can now open within versions 3 and above of Microsoft's *Internet Explorer* Web browser. Microsoft is intending to fully integrate Internet Explorer 4 with the next release of its immensely popular Windows operating system to the extent that the browser will become the interface which should ensure the future of this browser. *Visual Basic 5* also offers greater programming power than *SuperCard 3.0* and allows code created in other languages to be more easily used. In addition, media objects created in another application, for example Word text and equations, can

also be linked and embedded within *Visual Basic 5* and still remain easily editable in the application which created them.

9.6. Critique, reflection and the direction of future work

In some respects the project did not satisfy all of the original hopes for it. The project's collaborators in the School of Civil and Structural Engineering for example, were disappointed not to be given a neatly packaged CD ROM - based MLE describing the design of spillway terminal structures.

More seriously, although many of the ideas produced by the work can be seen to be generically relevant, and to have been described in those terms, this does not mean that the methods and techniques are comprehensive for the design of all types of MLE. Scope for further research exists applying the same analytical approach, rooted in cognitive science, to the design of MLEs in totally different types of domain. For example, interactive multimedia for rehearsing social and emotional issues in domains of learning such as childbirth (Jagodzinski et al 1995).

Even within the domain addressed, the research has not produced anything approaching a design method in the sense that, for example, SSADM provides a complete method for systems analysis. There is no detailed prescription of how a designer should apply an arsenal of specific techniques, stepwise, to the design process.

However, it can be argued that the field is not and may never be, ready for such a prescriptive method. The nature of human learning is still far from certain and rapid developments in neuroscience and connectionism would soon overtake any present attempts to make it concrete. Furthermore, learning, even as it is understood today, is too complex and varied to be comprehended by a single method.

What has become clear as a result of this work is the truth of the intuitive

suspicion expressed in section 1.1. that conventional approaches to the design of learning environments "are both incapable of capturing the richness of knowledge present in complex learning domains such as civil engineering and unable to take full advantage of the potential of the newer technology of computer based interactive multimedia" (page 7).

Chapter 3 in particular reviews a range of the design approaches which underlies much of the currently available interactive multimedia, and most of them have been criticised in the literature as having roots in the principles of human learning which are either too narrow or too shallow or both.

The intended contribution of the present work has been to show how it is possible to draw both in more breadth and in more depth on the principles of human learning and cognitive science in order to improve the design of multimedia learning environments. The preceding section has summarised the specific phases of the design process in which this contribution has been made. It would be possible to assemble a list of steps summarising, for each step, the fundamental cognitive principles, the techniques, the implementations and generic guidance which emerge from the work.

However, that would be to oversimplify the lessons which have been learnt and carried forward. A vital element which would be lost in such a list would be the sense of debate which has formed the most valuable part of the project. The collaborators in the work take from it not a list or prescriptive method but a sense of better understanding of the human cognitive issues and knowledge engineering techniques which, however imperfectly, provide the best we can do to design multimedia learning environments. The true value of the work is in the realisation that the design of MLEs needs the richest possible grasp of the competing and often contradictory messages from psychology, cognitive science, knowledge engineering and software design to date. The best manifestation of

this knowledge has taken the form of the design of university courses for designers of interactive multimedia in which the elements identified above are taught in the spirit of discursive debate and uncertainty which characterises human psychology. A discursive view of the world, in which uncertainty prevails, is usually alien to software engineers and designers and, initially, sometimes not welcomed by them. However, such courses are often seen in retrospect to provide some of the most valuable parts of degree programmes. The centre of gravity of computing is shifting away from business and technology to include more human-centred artifacts in education and entertainment. It is important that our theoretical bases and methodology should adapt to this shift.

In the immediate future the author intends, if circumstances permit, to implement the MLE further and transfer it to the World Wide Web, initially in *SuperCard 3.0* and then using *Visual Basic 5*, as discussed in the previous section. The author also intends to use ideas embodied in the CAM to inform the development of further computer based teaching material in related areas of Engineering. The author is currently a lecturer in Information Sciences at the University of Hertfordshire. In the 1997 - 98 academic year the Faculty of Information Sciences will be merging with the Faculty of Engineering and it is hoped the ensuing closer relationship will help the author to develop the work further. Members of the Faculty of Engineering are already involved in related work, for example, the IDER (Intelligent Design Engineering Research) project into concurrent engineering design using intelligent multimedia which, amongst other work, has led to the development of the ICE (Interactive Concurrent Engineering) multimedia computer aided learning software. Members of the group have already expressed an interest in the author's work.

Abbreviations used

CAL	-	Computer Aided Learning
CAM	-	Cognitive Apprenticeship Model
CBL	-	Computer Based Learning
CEng	-	Chartered Engineer
CPR	-	Chartered Professional Review
ICE	-	Institution of Civil Engineers
ID	-	Instructional Design
ISD	-	Instructional Systems Development
ITS	-	Intelligent Tutoring System
MICE	-	Corporate Membership of the Institution of Civil Engineers
MLE	-	Multimedia Learning Environment
SCE	-	Supervising Civil Engineer
USBR	-	United States Bureau of Reclamation

Appendix I: Solution to the elicited example design problem.

The original problem was taken from the “Design of Small Dams” (US Bureau of Reclamation (1987), pages 400 - 402) and was explained and reworked by the Journeyman.

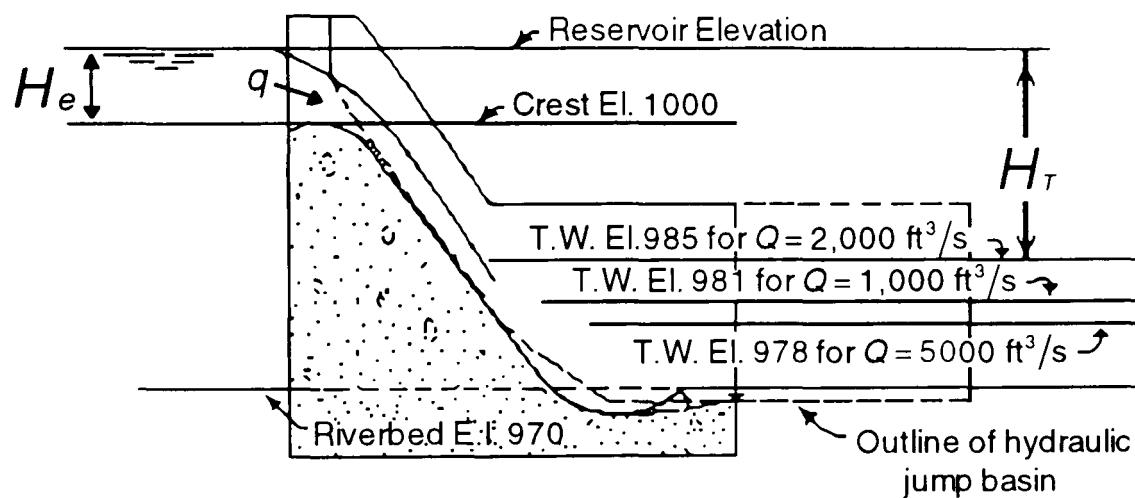
The solution outlined here is at a fairly rudimentary level and, for brevity, does not contain all the workings, thoughts and broader explanations elicited from the Journeyman. In accordance with this, some important areas in the solution have been annotated by “^(See MLE)” to indicate where further, in-depth background information, explanations, derivations and definitions were provided by the MLE.

Example designs of a stilling basin and an alternative submerged bucket dissipator.

An energy dissipation device is needed for the spillway of an overflow dam. Due to the prevailing site conditions it is thought worthwhile to produce comparative designs for both a hydraulic jump stilling basin and a submerged bucket dissipator.

The overflow dam's maximum discharge is 2,000 ft³/s and its controlling dimensions and tailwater conditions are shown in Figure 25.

Figure 25. Example design problem controlling dimensions and tailwater conditions (based on Figure 9-51, page 402, US Bureau of Reclamation (1987)).



For a first trial design, assuming a crest length of 20 feet, the criteria for a range of likely discharges are as follows:

Total discharge, Q , in	2,000	1,000	500
cubic feet per second			

Unit discharge, q , in	100	50	25
cubic feet per second	$\left(\frac{2,000}{20}\right)$	$\left(\frac{1,000}{20}\right)$	$\left(\frac{500}{20}\right)$

$$\text{per foot, } q = \left(\frac{Q}{\text{Crest Length}} \right)$$

Assumed coefficient of	3.9	3.7	3.5
discharge, C (See MLE)			

Head ^(See MLE) , in feet on crest,	8.7	5.7
	$H_e = \left(\frac{q}{C} \right)^{\frac{2}{3}}$	$\left[\left(\frac{100}{3.9} \right)^{\frac{2}{3}} \right] \quad \left[\left(\frac{50}{3.7} \right)^{\frac{2}{3}} \right] \quad \left[\left(\frac{25}{3.5} \right)^{\frac{2}{3}} \right]$

Reservoir water level	1008.7	1005.7
= crest elevation +	(1000 + 8.7)	(1000 + 5.7) (1000 + 3.7)
head on crest (H_e)		

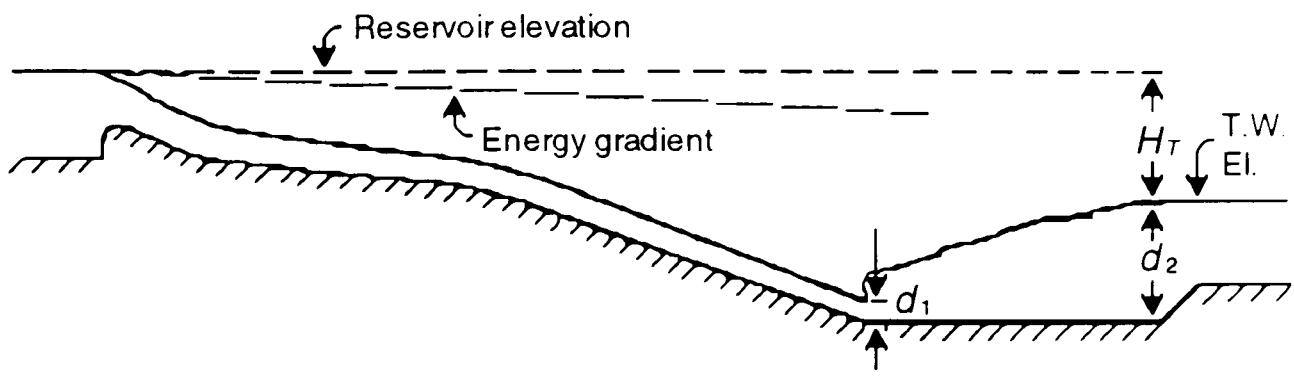
H_T , reservoir water	23.7	24.7
level minus tailwater	(1008.7 - 985)	(1005.7 - 981) (1003.7 - 978)
level		

So far, determining the general guiding criteria for the problem has been straightforward. Now, to produce a first trial design for a stilling basin involves:

- (i) fixing the basin's apron elevation
- (ii) selecting the most suitable type of stilling basin
- (iii) calculating the length of the basin

To determine the apron elevation the conjugate depth ^(See MLE) (d_2), after the hydraulic jump formed on the basin, is subtracted from the tailwater elevation (see Figure 26). This has to be done for each of the chosen discharges to determine the most critical condition.

Figure 26. Variables relating to the equation to determine stilling basin depths



Assuming “ α ” is the friction loss factor ^(See MLE) and “ g ” is the acceleration due to gravity (32.1538 ft/s²) then the following equation ^(See MLE) can be used to determine the conjugate depth (d_2):

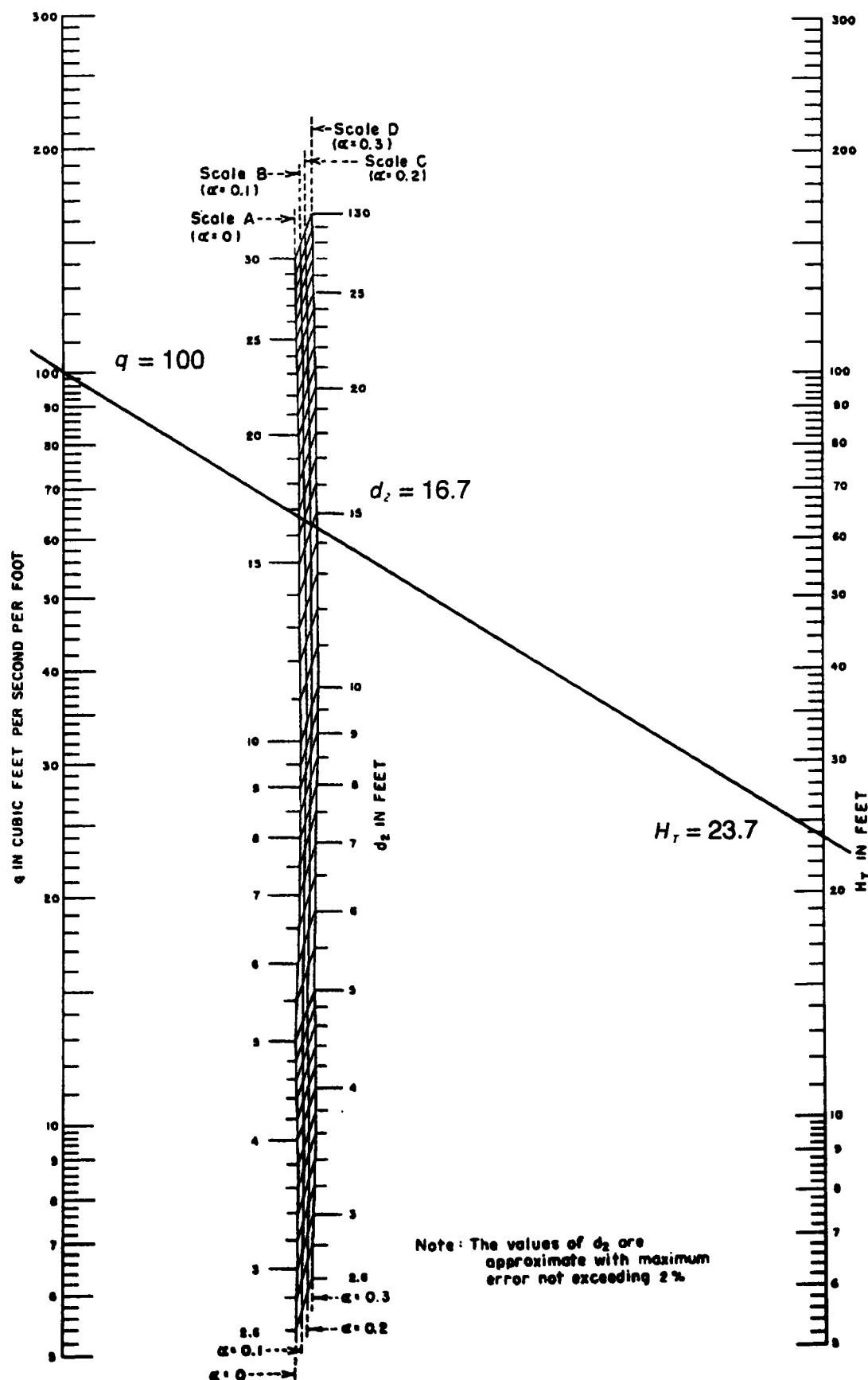
$$H_T = \frac{\frac{2q^2}{(1-\alpha)gd_2^2} \left[\sqrt{1 + \frac{8q^2}{gd_2^3}} - 1 \right]^2 + \frac{d_2}{2} \left(\sqrt{1 + \frac{8q^2}{gd_2^3}} - 1 \right)}{d_2}$$

The only unknown in the above equation is d_2 . Therefore, d_2 can be found by trial and error iteration. An easier, alternative method of determining d_2 would be to use an existing nomograph based on this equation, such as the one shown in Figure 27.

In Figure 27 a line has already been drawn on the nomograph for the first of the chosen range of discharges of $q = 100$ cubic feet per second per foot and its corresponding H_T of 23.7 feet. Assuming, for this design, a friction loss factor, α , of zero then for the above values d_2 can be read from the nomograph as 16.7

feet. Similarly, d_2 can be determined for the other two values in the range of chosen discharges.

Figure 27. Nomograph used to determine stilling basin depths (based on Figure 9-43, page 396, US Bureau of Reclamation (1987)).



So, in summary, determining d_2 and calculating the most suitable apron elevations for each of the discharges:

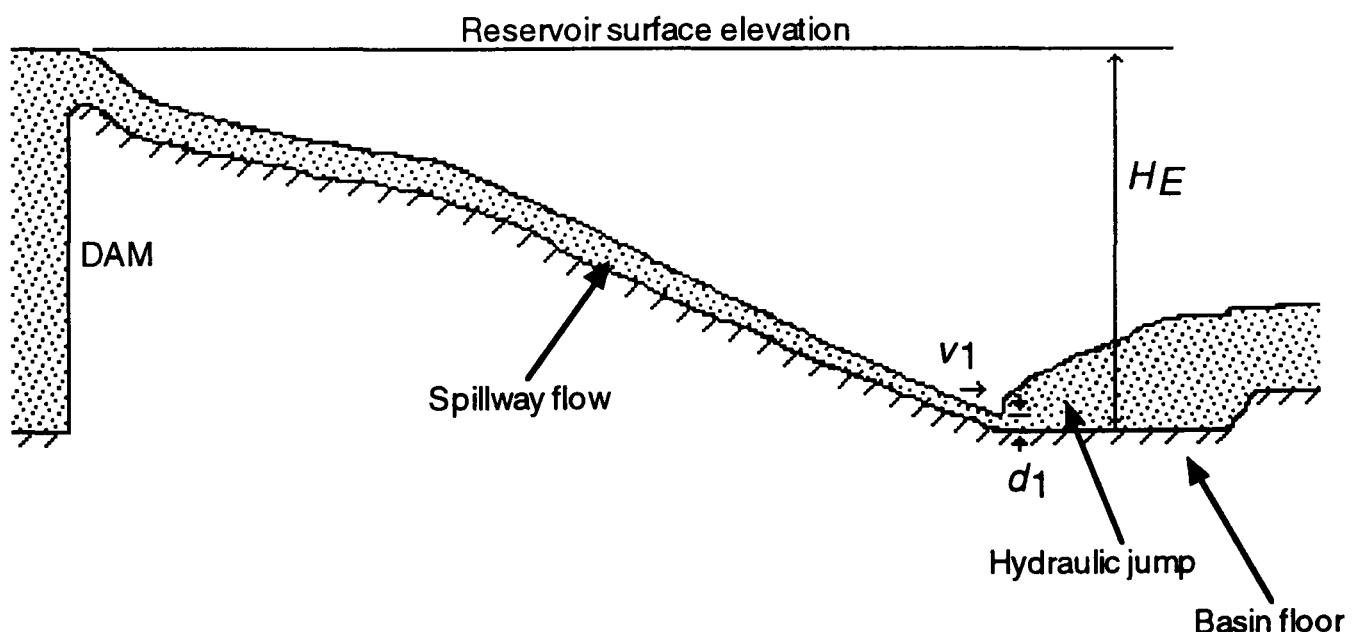
Unit discharge, q	100 ($Q = 2,000$)	50 ($Q = 1,000$)	25 ($Q = 500$)
H_T	23.7	24.7	25.7
Conjugate depth, d_2	16.7	11.8	8.6
Tailwater elevation	985.0	981.0	978.0
in feet (provided in question, otherwise tailwater rating curve nomographs <small>(See MLE)</small> can be used)			
Therefore, required apron elevation = tailwater elevation - conjugate depth, d_2	968.3 (985 - 16.7)	969.2 (981 - 11.8)	969.4 (978 - 8.6)

The lowest apron elevation of 968.3 ft for the largest discharge of 2,000 ft³/s is the most critical condition, so, the basin elevation will have to be 968.3 ft.

To select the most suitable type of hydraulic jump stilling basin the velocity of the water entering the basin needs to be considered, v_1 , (see Figure 28) together with the form of the hydraulic jump which will develop on the basin (See MLE). The

different forms of hydraulic jump which can develop have been shown to be related to the Froude number parameter ^(See MLE): $F_1 = \frac{V_1}{\sqrt{g d_1}}$ (see stills from animation, Figure 16, Chapter 5).

Figure 28. Variables used to help select the required type of stilling basin.



To find the incoming velocity, V_1 , the equation to find specific energy, H_E , can be used ^(See MLE):

$$H_E = d_1 + \frac{V_1^2}{2g}$$

H_E (at upstream end of basin)

= reservoir water surface minus apron elevation, assuming no loss in specific energy

$$= 1008.7 - 978.3 = 40.4$$

To find the upstream depth of flow at basin floor level, d_1 , the hydraulic jump equation can be used ^(See MLE).

$$d_1 = \frac{1}{2} d_2 \left(-1 + \sqrt{1 + \frac{8q^2}{g d_2^3}} \right)$$

$$d_1 = \frac{1}{2} \times 16.7 \left(-1 + \sqrt{1 + \frac{8(100)^2}{32.1538 \times (16.7)^3}} \right)$$

$$\therefore d_1 = 2 \text{ ft}$$

Rearranging the specific energy equation and substituting in the values of H_E

and d_1

$$V_1 = \sqrt{2g (H_E - d_1)}$$

$$V_1 = \sqrt{2 \times 32.1538 (40.4 - 2)}$$

$$\therefore V_1 = 49.7 \text{ ft/s}$$

The Froude number can now be calculated using the equation:

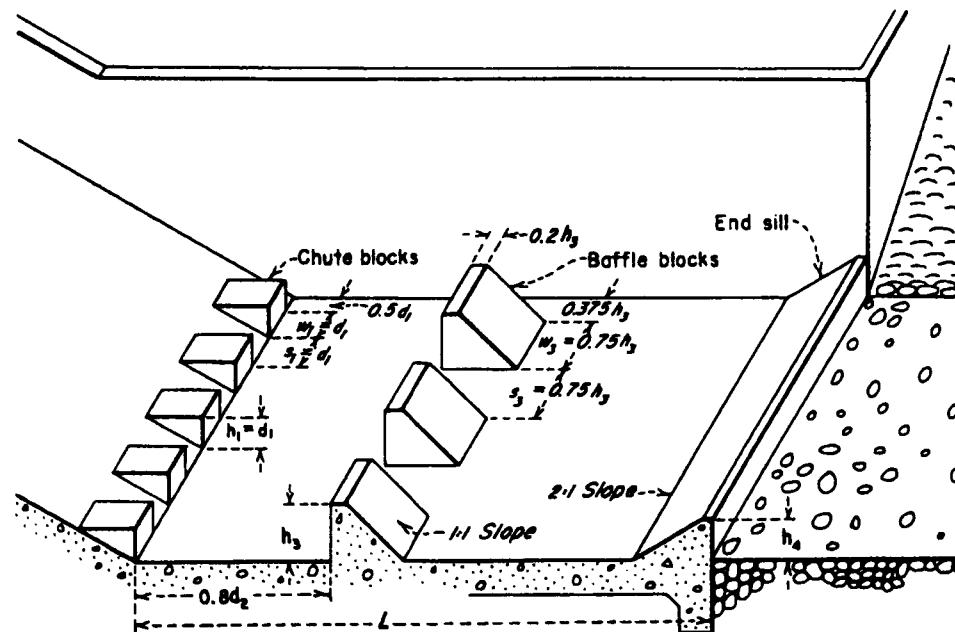
$$F_1 = \frac{V_1}{\sqrt{g d_1}}$$

$$F_1 = \frac{49.7}{\sqrt{32.1538 \times 2}}$$

$$\therefore F_1 = 6.2$$

The most appropriate type of hydraulic jump stilling basin to use is determined by considering the incoming velocity, v_1 , and the Froude number, F_1 . In this situation, a US Bureau of Reclamation type III stilling basin is recommended (see Figure 29) because the Froude number is more than 4.5 and the incoming velocity does not exceed 60 ft/s.

Figure 29. Dimensions for a type III hydraulic jump stilling basin (taken from Figure 9-41, page 393, US Bureau of Reclamation (1987)).



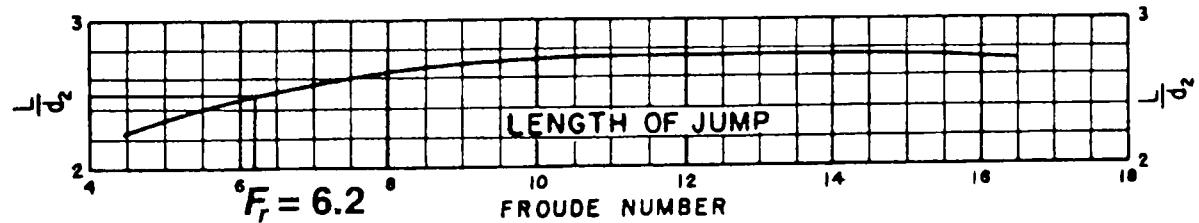
As the Froude number is more than 4.5 a true hydraulic jump will form (See MLE).

The type III basin uses a number of accessory devices: chute blocks, impact baffle blocks and an end sill to provide a stabilising effect on the jump and allow the basin length to be shortened when compared to, for example, a type I basin (see Figure 3, Chapter 2) which uses no accessory devices. The accessory devices also provide a safety factor against sweepout caused by inadequate tailwater depth.

The type III basin relies on dissipation of energy by the impact blocks and on the turbulence of the hydraulic jump ^(See MLE) phenomena (see video still, Figure 4, Chapter 2) for its effectiveness. Because of the large impact forces to which the baffles are subjected by the high incoming velocities and because of the possibility of cavitation ^(See MLE) along the surfaces of the blocks and floor, the use of this basin is limited to heads where the incoming velocity does not exceed 60 ft/s.

A nomograph such as the one in Figure 30 can be used to fix the length of a type III stilling basin.

Figure 30. Nomograph used to fix the length of a US Bureau of Reclamation type III stilling basin (based on Figure 9-41, page 393, US Bureau of Reclamation (1987)).

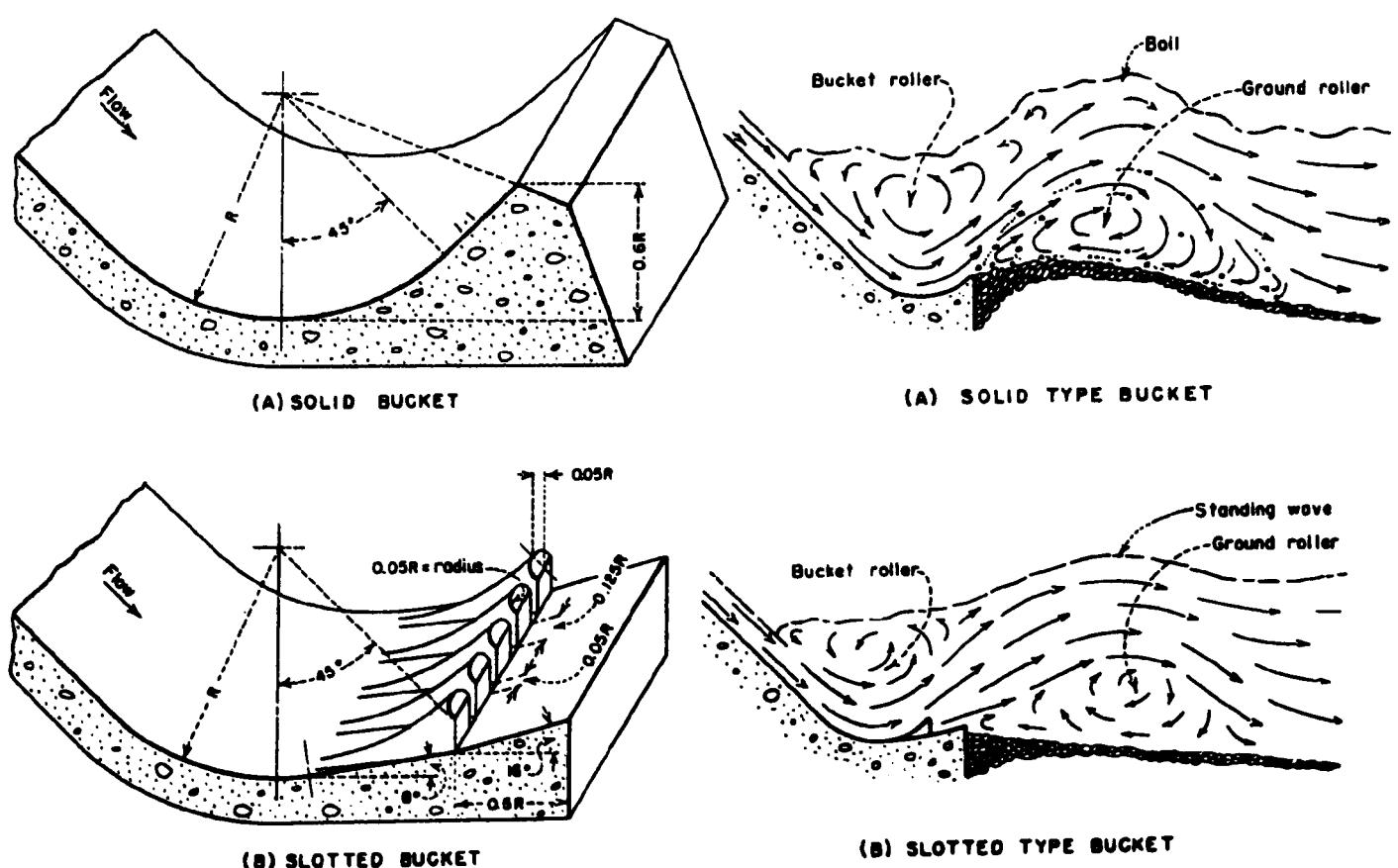


The nomograph matches Froude number to $\frac{L}{d_2}$. d_2 is already known so L , length of the stilling basin, can be found. Lines have been drawn on the nomograph for the Froude number of 6.2. to give a reading for $\frac{L}{d_2}$ of 2.5. So, given that $d_2 = 16.7$ then the length of the type III stilling basin should be $2.5 \times 16.7 = 42$ feet .

This problem also allows for the use of a submerged or roller bucket dissipator. Although submerged bucket dissipators are most often used when the application of a stilling basin is prohibited because the tailwater depth is too great for the formation of a hydraulic jump.

Two types of submerged buckets, solid and slotted, have been developed and model tested by the US Bureau of Reclamation (see the left hand side of Figure 31).

Figure 31. Submerged buckets and their hydraulic action (taken from Figures 9-45 and 9-46, pages 398 and 399, US Bureau of Reclamation (1987)).

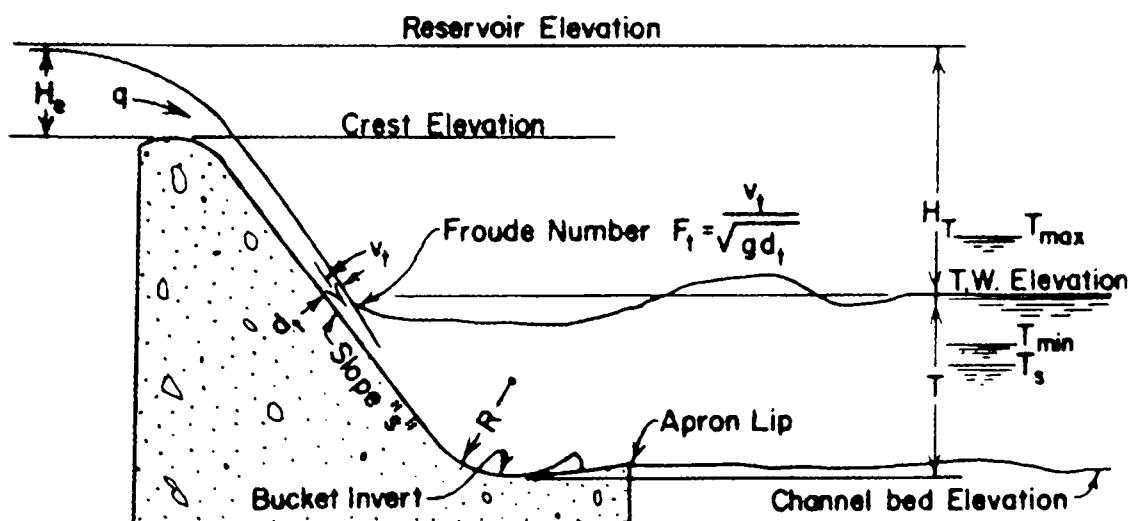


The general nature of the dissipating action for each type of bucket is the same. Both buckets cause the creation of an upstream, surface anti-clockwise roller and a downstream, ground, clockwise roller (see the right hand side of Figure 31).

However, distinctive features of the flows produced by each bucket differ to the extent that each bucket can be seen to possess certain limitations. The slotted bucket provides better energy dissipation with less severe surface and streambed disturbances. However, it is more sensitive to sweepout at lower tailwaters and is conducive to a diving and scouring action at excessive tailwaters (See MLE). This is not the case with the solid bucket. Thus, the tailwater range that provides good performance with the slotted bucket is much narrower than that of the solid bucket. A solid bucket dissipator should not be used where the tailwater limitations of the slotted bucket can be met.

Initially, for this problem, the design of a slotted bucket will be considered to determine whether it can meet the given tailwater variations.

Figure 32. Variables used in the design of a submerged bucket dissipator (taken from Figure 9-50, page 402, US Bureau of Reclamation (1987)).



Using the variables in Figure 32 and some of the variables and values previously calculated from Figure 25:

Total discharge, Q	2,000	1,000	500
--------------------	-------	-------	-----

Unit discharge, q	100	50	25
H_T , reservoir water level minus tailwater level	23.7	24.7	25.7
Velocity head at tailwater level, h_{V_t} , in feet (assuming no loss of specific energy)	23.7	24.7	25.7
Velocity of flow, in feet per second at tailwater level, $V_t = \sqrt{2g h_{V_t}}$	39.1	39.9	40.7
Depth of flow, in feet, at tailwater level, $d_t = q/V_t$	2.56	1.25	0.61
Froude number at tailwater level, $F_t = V_t/\sqrt{gd_t}$	4.3	6.3	9.2
Specific energy at tailwater level, $d_t + h_{V_t}$	26.3	25.9	26.3

For the design of the submerged slotted bucket, the minimum bucket radius for the maximum discharge is determined by the use of Figure 33. With a Froude number at tailwater level of $F_r = 4.3$, the minimum radius is $0.42(d_t + h_{v,t}) = 0.42 \times 26.3 = 11$ feet.

In this problem the riverbed slopes up. Using Figure 33 produces the following values for the minimum and maximum tailwater levels when $F_r = 4.3$ and $R/(d_t + h_{v,t}) = 0.42$:

$$T_{\max} = 7.5 d_t = 7.5 \times 2.56 = 19.2 \text{ feet}$$

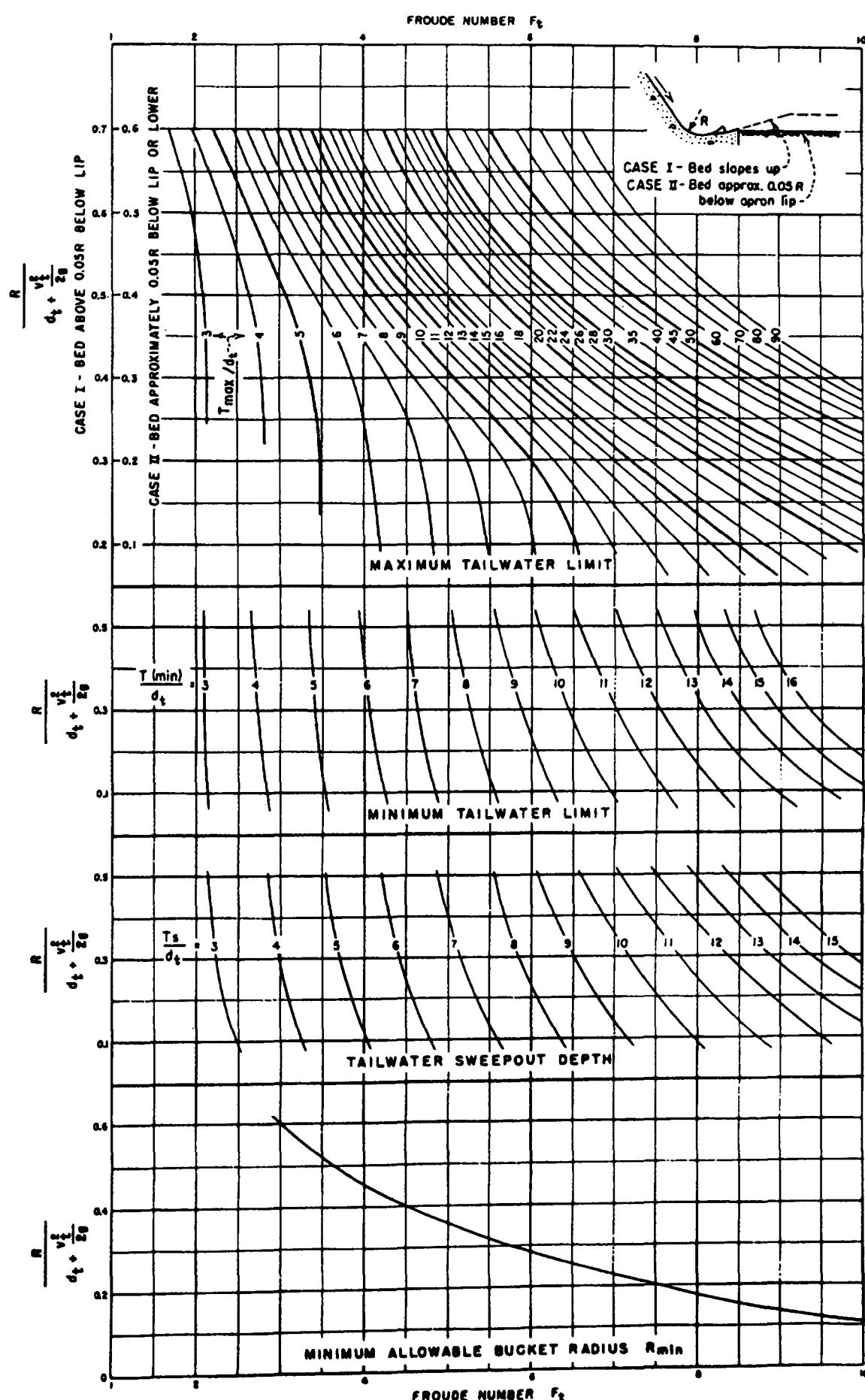
$$T_{\min} = 6.5 d_t = 6.5 \times 2.56 = 16.6 \text{ feet}$$

So, an average tailwater depth of, approximately, 18 feet will result in the inverted bucket having to be positioned at an elevation of $985 - 18 = 967$ feet.

Now, the radius and tailwater conditions for the other, less than maximum flows, have to be determined to ensure the design is satisfactory throughout the range of discharge.

For the total discharge of $1,000 \text{ ft}^3/\text{s}$ with a Froude number at tailwater level of $F_r = 6.3$, using Figure 33 and values already calculated, the minimum radius

Figure 33. Nomograph for the design of a slotted submerged bucket dissipator
 (taken from Figure 9-49, page 401, US Bureau of reclamation (1987)).



is given as $0.26(d_t + h_{v_i}) = 0.26 \times 25.9 = 6.8$ feet. This means the minimum radius of 11 feet determined for the maximum discharge of $2,000 \text{ ft}^3/\text{s}$ will

govern. The maximum and minimum tailwater values for $F_r = 6.3$ and $R/(d_r + h_{v_r}) = 0.42$ from Figure 33 are:

$$T_{\max} = 20 d_r = 20 \times 1.25 = 25 \text{ feet}$$

$$T_{\min} = 10.1 d_r = 10.1 \times 1.25 = 12.6 \text{ feet}$$

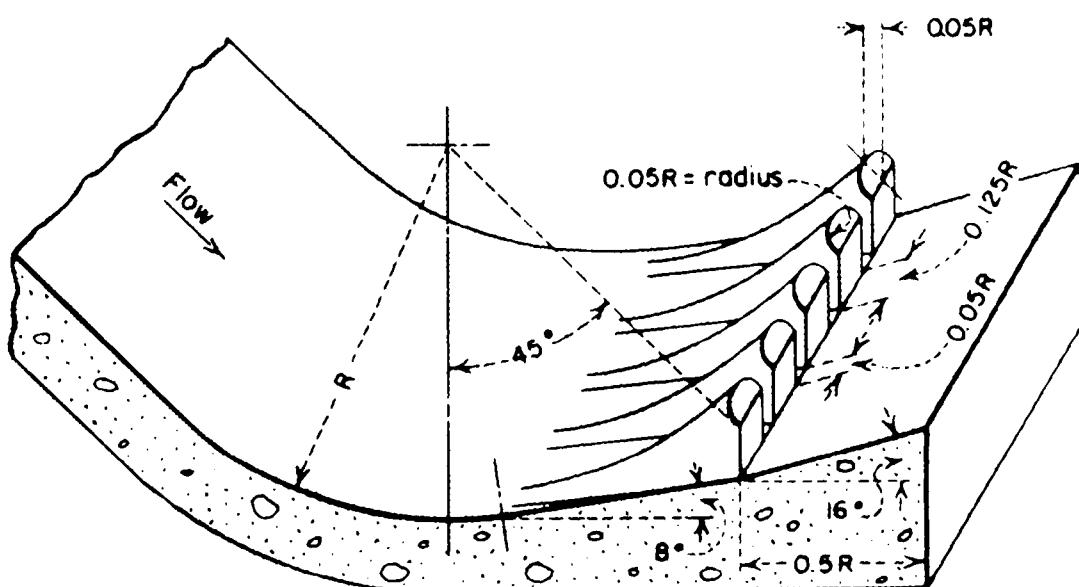
The bucket elevation of 967 calculated for the maximum discharge of $2,000 \text{ ft}^3/\text{s}$ will produce a tailwater depth of $981 - 967 = 14$ feet which is within the safe limit for providing satisfactory roller action.

The same steps are also used to guarantee satisfactory roller action will occur with the minimum total discharge of $500 \text{ ft}^3/\text{s}$. With this discharge the minimum radius of 11 feet, determined for the maximum discharge, only just proved to be acceptable. T_{\max} and T_{\min} were found to be 50 feet and 10.4 feet, respectively, compared with the 11 feet of tailwater depth provided by the bucket elevation of 967 feet.

Although the design based on the maximum discharge can be said to be satisfactory for the lower discharges, if a wider range of safe tailwater depths is desired, the radius of curvature of the bucket can be increased. So, more leeway for tailwater variations will be provided by a bucket radius of 12 feet. For this radius the maximum discharge, $T_{\min} = 6.5 d_r = 16.6$ feet and $T_{\max} = 8.5 d_r = 22.5$ feet. Then an average tailwater depth of, approximately, 20 feet will place the bucket at an elevation of $(985 - 200) = 965$ feet. So, a submerged slotted

bucket is also appropriate for the given conditions and it will have to be dimensioned as shown in Figure 34.

Figure 34. Dimensions for a submerged slotted bucket dissipator (taken from Figure 9-45, page 398, US Bureau of Reclamation (1987)).



References

- Adelman, L. (1987) 'Toward considering psychological measurement issues when developing expert systems', *Proceedings of the 1987 IEEE International Conference on Systems, Man and Cybernetics (Cat. No. 87CH2503 - 1)*, Alexandria, VA, USA, 20 - 23 October 1987 (Piscataway, NJ: IEEE), **2**, pp. 1044 - 1048.
- Anderson, J.R. & Jeffries, R. (1985) 'Novice LISP errors: Undetected losses of information from working memory', *Human-Computer Interaction*, **22**, pp. 403 - 423.
- Anderson, J.R. (1991) 'Is human cognition adaptive?', *The Behavioural and Brain Sciences*, **14**, pp. 471 - 484.
- Anderson, J.R. (1993) *Rules of the Mind*, Lawrence Erlbaum Associates Inc Publishers, Hillsdale, New Jersey.
- Andrews, D.H., & Goodson, L.A. (1980) 'A comparative analysis of models of instructional design', *Journal of Instructional Development*, **3**(4), pp. 2 - 16.
- Bainbridge, L. (1979) 'Verbal reports as evidence of the process operator's knowledge', *International Journal of Man - Machine Studies*, **11**, pp. 411 - 436.
- Baker, K., Ball, L., Culverhouse, F. F., Dennis, I., Evans, J., Jagodzinski A.P., Pearce, P.D., Scothern, D. & Venner, G. (1991) 'A psychologically based intelligent design aid', In: *Intelligent CAD systems III*, Springer Verlag, Berlin.
- Baker, L. & Brown, A.L. (1980) 'Metacognitive skills of reading', In: Pearson, D. (ed), *Handbook of reading research*, Longman, New York, pp. 353 - 394.
- Ball, L.J. (1990) *Cognitive processes in engineering design*, PhD Thesis, University of Plymouth.

Barden, R. (1989) 'Developing a HyperCard-based intelligent training system', *Educational & Training Technology International*, **26** (4), pp. 361 - 367.

Barrese, R.M., Calabró, G., Cozza, S., Gallo, T., & Tisato, F. (1992) 'CAMCE: An environment to support multimedia courseware projects', *Educational & Training Technology International*, **29** (1), pp. 14 - 25.

Bechtel, W. & Abrahamsen, A. (1991) *Connectionism and the mind*, Blackwell, Cambridge, Mass.

Becker, L.A. (1988) 'Computer-aided misconception-based intelligent tutoring and exercise generation', *Programmed Learning and Educational Technology*, **25** (1) , pp. 67 - 72.

Bereiter, C. (1991) 'Implications of connectionism for thinking about rules', *Educational Researcher*, April, Vol. 20, No. 3, pp. 10 - 16.

Berners-Lee, T., Cailliau, R., Luotonen, A. Nielsen, F. & Secret, A. (1994), 'The World-Wide Web', *Communications of the ACM*, **37**(8), pp. 76 - 82.

Bigelow, J. (1988) 'Hypertext and CASE', *IEEE COMPUTER*, pp. 23 - 27, March.

Bobrow, D.G. (ed) (1985) *Qualitative reasoning about physical systems*, MIT Press, Cambridge, MA.

Bobrow, D.G., Mittal, S. & Stefik, M.J. (1986) 'Expert systems: perils and promises', *Communications of the ACM*, **29** (9), pp. 880 - 894.

Bolt, R., (1979) *Spatial Data-Management*, Architecture Machine Group, M.I.T. Cambridge, Mass.

Bonar, J.G. & Cunningham, R. (1988) 'Bridge: Tutoring the programming process', In: Psotka, J., Massey, L.D., & Mutter, S.A. (eds), *Intelligent Tutoring Systems: Lessons Learned*, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 409 - 434.

Bork, A. (1984) 'Producing computer based learning material at the educational technology centre', *Journal of Computer Based Instruction*, 11(3), pp. 78 - 81.

Boyle, J.M. (1989) 'Interactive engineering systems design: a study for artificial intelligence applications'

Braden, R.A. (1996) 'The case for linear instructional design and development: A commentary on models, challenges, and myths', *Educational Technology*, 36(2), March - April, pp. 5 - 23.

Bransford, J.D., Goin, L.I., Hasselbring, T.S., Kinzer, C.K., Sherwood, R.D., & Williams, S. (1988) 'Learning with technology: Theoretical and empirical perspectives', *Peabody Journal of education*, 64(1), pp. 5 - 26.

Branson, R.K. & Grow, G. (1987) 'Instructional systems development', In: Gagné, R.M. (ed), *Instructional technology; Foundations*, Lawrence Erlbaum, Hillsdale, NJ.

Breuker, J.A. & Wielinga, B.J. (1984) 'Techniques for knowledge elicitation and analysis', *Report 1.5, Esprit Project 12 (Memorandum 28 of the Research Project "The Acquisition of Expertise")*, July 1984, Amsterdam.

Breuker, J.A. & Wielinga, B.J. (1987) 'Use of models in the interpretation of verbal data', In: *Knowledge Acquisition for Expert Systems: A Practical Handbook*, Kidd, A.L. (Ed.), NY: Plenum, New York, pp. 17 - 44.

Brown, P.J. (1991) 'Hypertext: Dreams and Reality. In: Brown, H. (ed), *Hypermedia/Hypertext: An object-oriented Database*, Chapman and Hall, London.

Brown, J.S. (1985a) 'Process versus product; A perspective on tools for communal and informal electronic learning', *Journal of Educational Computing Research*, 1, pp. 179 - 201.

Brown, J.S. (1985b) 'Idea-amplifiers: New kinds of electronic learning', *Educational Horizons*, 63, pp. 108 - 112.

Brown, J.S. & Burton, R.E. (1978) 'Diagnostic models for procedural bugs in basic mathematical skills', *Cognitive Science*, **2**, pp. 155 - 192.

Brown, J.S., Burton, R.R. & de Kleer, J. (1982) 'Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III', In: Sleeman, D.H. & Brown, J.S. (eds), *Intelligent tutoring systems*, Academic Press, London, pp. 227 - 282.

Brown, J.S., Collins, A. & Duguid, P. (1989), 'Situated cognition and the culture of learning', *Educational Researcher*, January - February, pp. 32 - 42.

Bruner, J.S. (1966) *Towards a Theory of Instruction*, The Belknap Press of Harvard University Press: Cambridge MA. and London.

Burke, R.L. (1982) *CAI sourcebook*, Prentice Hall, New Jersey.

Burns, H. L. & Capps, C.G. (1988) 'Foundations of intelligent tutoring systems: an introduction', In: Polson, M.C. & Richardson, J.J. (eds), *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum, London, pp. 1 - 19.

Bush, V. (1945) 'As we may think', *Atlantic Monthly*, July, pp. 101 - 108.

Cennamo, K.S., Savenye, W.C. & Smith, P.L. (1991) 'Mental effort and video-based learning: The relationship of preconceptions and the effects of interactive and covert practice', *Educational Technology Research and Development*, **39** (1), pp. 5 - 16.

Chi, M.T.H., Bassok, M., Lewis, M., Reimann, P. & Glaser, R. (1989) 'Self-explanations: How students study and use examples in learning to solve problems', *Cognitive Science*, **13**, pp. 145 - 182.

Chi, M.T.H., Glaser, R. & Farr, M.J. (1988) *The nature of expertise*, Lawrence Erlbaum, Hillsdale, NJ.

Clancey, W.J. (1984) 'Methodology for building an intelligent tutoring system', In: Kintsch, W., Miller, J.R. & Polson, P.G. (eds), *Methods and Tactics in Cognitive Science*, Lawrence Erlbaum, London.

Clancey, W.J. (1986) 'From GUIDON to NEOMYCIN and HERACLES in twenty short lessons: ONR final report, 1979-1985, *AI Magazine*, 7(3), pp. 40 - 60.

Collins, A. (1991) 'Cognitive apprenticeship and instructional technology', In: Idol, L. & Jones, B.F. (eds), *Educational values and cognitive instruction: implications for reform*, Lawrence Erlbaum Associates, New Jersey.

Collins, A. & Brown, J.S. (1988) 'The computer as a tool for learning through reflection', In: Mandl, H. & Lesgold, A. (eds), *Learning issues for intelligent tutoring systems*, Springer-Verlag, New York, pp. 1 - 18.

Collins, A., Brown, J.S. & Newman, S.E. (1989) 'Cognitive apprenticeship: teaching the crafts of reading, writing and mathematics', In: Resnick, B. (ed), *Essays in honour of Robert Glaser*, Lawrence Erlbaum, Hillsdale, N.J.

Collins, A. & Smith, E.E. (1982) 'Teaching the process of reading comprehension', In: Detterman, D.K. & Sternberg, R.J. (eds) *How much and how can intelligence be increased?*, Ablex, Norwood, NJ, pp. 173 - 185.

Collins, A. & Stevens, A.L. (1982) 'Goals and strategies of inquiry teachers', In: Glaser, R. (ed), *Advances in instructional psychology*, Volume 2, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 65 - 119.

Collins, A. & Stevens, A.L. (1983) 'A cognitive theory of interactive teaching', In: Reigeluth, C.M. (ed), *Instructional-design theories and models: An overview of their current status*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Conklin, J. (1987) 'Hypertext: An introduction and survey', *IEEE Computer*, 20(9), pp. 17 - 41.

Corbett, A.T. & Anderson, J.R. (1990), 'The effect of feedback control on learning to program with the LISP tutor', *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, pp. 796 - 803.

Creanor, L., Durndell H. & Primrose, C. (1996) 'Library and study skills using hypertext: the TILT experience', *New Review of Hypermedia and Multimedia*, **2**, pp. 121 - 148.

Crowder, N.A. (1959) 'Automatic tutoring by means of intrinsic programming', In: Galanter, E. (ed), *Automatic Teaching: the State of the Art*, Wiley, New York.

Crowder, R., Hall, W., Heath, I., & Bernard, R. (1995) 'The application of large-scale hypermedia information systems to training', *Innovations in Education and Training International*, **32**(3), pp. 245 - 255.

Csikszentmihalyi, M. (1982) 'Towards a psychology of optimal experience', In: Wheeler, L. (ed), *Review of Personality and Social Psychology*, Volume 2, Beverly Hills CA, Sage, pp. 13 - 36.

Cybulski, J.L. & Reed, K. (1992) 'A hypertext based software engineering environment', *IEEE Software*, **9** (2), pp. 62 - 68.

Davis, H., Hall, W., Heath, I., Hill, G., & Wilkins, R. (1992) 'Towards an integrated information environment with open hypermedia systems', *Proc. ECHT'92 Fourth ACM Hypertext Conf.*, pp. 181 - 190.

Davis, H. C., Knight, S., & Hall, W. (1994) 'Light hypermedia link services: A study of third party application integration', *Proc. ACM ECHT'94 European Conference on Hypermedia Technology*, Edinburgh, 18 - 23 September, pp. 41 - 50.

De Corte, E. (1990) 'Learning with new information technologies in schools: perspectives from the psychology of learning and instruction', *Journal of Computer Assisted Learning*, **6**, pp. 69 - 87.

Delisle, N. & Schwartz, M. (1987) 'Contexts - A partitioning concept for hypertext', *ACM Trans. Office Information Systems* **5**(2), April, pp. 168 - 186.

Desforges, C. (1989) 'Understanding learning for teaching', *Westminster Studies in Education*, Vol. 12, pp. 17 - 29.

Dick, W. & Carey, L.M. (1990) *The systematic design of instruction* (3rd ed.), Harper Collins Publishers, Glenview, IL.

Dixon, J.R. (1987) 'On research methodology towards a scientific theory of engineering design', *Artificial Intelligence for Engineering Design. Analysis and Manufacturing*, 1 (3), pp. 145 - 157.

Duffield, J.A. (1991) 'Designing computer software for problem-solving instruction', *Educational Technology Research and Development*, 39 (1), pp. 50 - 62.

Duffy, T.M. & Jonassen, D.H. (1992) *Constructivism and the technology of instruction: A conversation*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Dunn, T.G. & Taylor, C.A. (1990) 'Hierarchical structures in expert performance', *Educational Technology Research and Development*, 38 (2), pp. 5 - 18.

Dym, C.L. & Levitt, R.E. (1991) *Knowledge based systems in engineering*, McGraw-Hill, Singapore.

Elavatorski E .A. (1959) *Hydraulic Energy Dissipators*, McGraw Hill, New York.

Elsom-Cook, M. (1987) *Intelligent Computer-aided instruction research at the Open University*. Technical Report No.: 63. Computer-Assisted Learning Research Group, The Open University, Milton Keynes.

Ericsson, K.A. & Simon, H.A. (1984) *Protocol Analysis: Verbal reports as data*, The MIT Press, Cambridge, MA.

Fidoro, J. (1988) 'A grand vision', *Byte*, 100, pp. 237 - 244.

Gagné, R.M. (1966) *The conditions of learning* (1st ed.), Holt, Rinehart and Winston, New York.

Gagné, R.M. (1985) *The conditions of learning* (14th ed.), Holt, Rinehart and Winston, New York.

Haake, A., Hüser, C., & Reichenberger, K. (1994) 'The individualized electronic newspaper: An example of an active publication', *Electronic Publishing: Origination, Dissemination and Design (EP-ODD)*, special issue on active documents.

Haertel, H. (1987) *A qualitative approach to electricity*, Palo Alto, CA, Xerox Corporation, Institute for Research on Learning.

Halasz, F.G., Moran, T.P., & Trigg, R.H. (1987) 'NoteCards in a nutshell', *Proc. ACM CHI&GI'87*, Toronto, Canada, 5 - 9 April, pp. 45 - 52.

Hall, W. (1996) 'Creating the perfect computer assistant', *EPSRC Newsline*, July, Issue 7, pp. 20 - 21.

Hall, W., & Davis, H. (1994) 'Hypermedia link services and their application to multimedia information management', *Information and Software Technology*, 36 (4), pp. 197 - 202.

Hall, W., & Woolf, B.P. (1995) 'Multimedia Pedagogues: Interactive Systems for Teaching and Learning', *Computer*, May.

Hall, W., Thorogood, Hutchings, G., & Carr, L. (1989). Using HyperCard and interactive video in education: An application in cell biology. *Educational & Training Technology International*, 26 (3), pp. 207 - 210.

Hammond, N. (1991) 'Tailoring Hypertext for the Learner', In: Kommers, P.A.M., Jonassen, D.H. & Mayes, J.T. (eds), *Cognitive Tools for Learning*, Springer-Verlag, Berlin, pp. 149 - 160.

Hartley, J.R. & Sleeman, D.H. (1973) 'Towards more intelligent teaching systems', *International Journal of Man-Machine Studies*, 5, pp. 215 - 236.

Hayes, J.R., & Flower, L.S. (1980) 'Identifying the organization of writing processes', In: Gregg, L.W. & Sternberg E.R. (eds), *Cognitive processes in writing*, Lawrence Erlbaum, Hillsdale, NJ.

Heath, I., Hall, W., Crowder, R.W., Pasha, M.A. & Soper, P.J. (1994) 'Integrating a knowledge base with an open hypermedia system and its application in an industrial environment', *Proceedings of the 3rd International Conference on Information and Knowledge Management CIKM '94 Workshop on Intelligent Hypertext*, Maryland, NIST Gaithersburg.

Hill, G., and Hall, W. (1994) 'Extending the Microcosm model to a distributed environment', *Proc. ACM ECHT'94 European Conference on Hypermedia Technology*, Edinburgh, U.K., 18 - 23 September, pp. 32 - 40.

Hollan, J.D., Hutchins, E.L. & Weitzman, L. (1984) 'STEAMER: An interactive inspectable simulation-based training system', *AI Magazine*, 5, pp. 15 - 27.

Jackson, P. (1990) *Introduction to expert systems*, Addison Wesley, Wokingham, England.

Jacobs, G. (1992) 'Hypermedia and discovery-based learning: a historical perspective', *British Journal of Educational Technology*, 23 (2), pp. 113 - 121.

Jagodzinski, P., Parmee, I. & Smith, C. (1994) 'The use of multiple paradigms for the representation of knowledge in a multimedia learning environment for engineering design', *Paper submitted to the International Journal of Human-Computer Studies*.

Jagodzinski, A. P., Phillips, M.G., Rogers, T. & Smith, C. (1995) 'Models of knowledge, learning and representation for multimedia learning environments', *Association for Education and Training Technology International Conference*, University of Plymouth, April 10-12 1995.

Jefferies, R., Turner, A.A., Polson, P.G. & Atwood, M.E. (1981) 'The processes involved in designing software', In: Anderson, J.R. (ed), *Cognitive skills and their Acquisition*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Johnson-Laird, P.N. (1983) *Mental models: Toward a cognitive science of language, inference and consciousness*, Cambridge University Press, Cambridge, England.

Jonassen, D.H. (1987) 'Verifying a method for assessing cognitive structure using pattern notes', *Journal of Research and Development in Education*, **20**(3), pp. 1 - 14.

Jonassen, D.H. (1990) 'Semantic network elicitation: Tools for structuring hypertext', In: McAleese, R. & Green, C. (Eds.), *Hypertext: The state of the art*, Intellect, London.

Jones, M.K. (reviews the work of Marc Meurrens and his colleagues at the Free University of Brussels), (1990). G: An object oriented learning system. *Educational Technology Research and Development*, **38** (2), pp. 93 - 97.

Kahn, G., Nowlan, S. & McDermott, J. (1985) 'Strategies for knowledge acquisition', *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-7*(5), pp. 511 - 522.

Koper, R. (1995) 'PROFIL: a method for the development of multimedia courseware', *British Journal of Educational Technology*, **26**(2), pp. 94 - 108.

Kozma, R.B. (1991) 'Learning with media', *Review of Educational Research*, **61**(2), pp. 179 - 211.

Kuipers, B.J. & Kassirer, J.P. (1984) 'Causal reasoning in medicine: analysis of protocol', *Cognitive Science*, **8**, pp. 363 - 385.

Lakoff, G. (1987) *Women, fire, and dangerous things: What categories reveal about the mind*, Chicago University Press, Chicago, IL.

Landow, G.P. 'Hypertext in Literary Education, Criticism, and Scholarship', *Department of English/Institute for Research in Information and Scholarship (IRIS)*, Providence, Rhode Island, Brown University.

Landow, G.P. & Kahn, P. (1992) 'Where's the hypertext? The Dickens Web as a system-independent hypertext', *Proc. ACM ECHT'92 Conf.*, Milan, Italy, 30 November - 4 December, pp. 149 - 160.

Laurillard, D. (1991a) 'How Computers Assist Learning: Unit 2', Block 1: Learning Through Computers, EH232 Education/Educational Technology: a second level course, The Open University, Milton Keynes.

Laurillard, D. (1991b) *Learning through collaborative computer simulations*, Centre for Information Technology in Education report No. 18, 13/12/1991.

Lencastre, A. (1987) *Handbook of hydraulic engineering*, Ellis Horwood Limited, Chichester.

Lindsay, R. & Norman, D.A. (1977) *Human information processing*, Academic Press, New York.

Locatis, C., Charuhas, J. & Banvard, R. (1990) 'Hypervideo', *Educational & Training Technology International*, **38** (2), pp. 41 - 49.

MacDonald, B. & the UNCAL team members (1976) 'The educational potential of computer-assisted learning: qualitative evidence about student learning', paper prepared by UNCAL, the independent educational evaluation of the NDPCAL, CARE, UEA.

Mack, R.L. & Nielsen, J. (eds) (1994) *Usability Inspection Methods*, John Wiley & Sons, New York, NY.

Mandl, H. & Lesgold, A. (eds) (1988) *Learning Issues for Intelligent Tutoring Systems*, Springer-Verlag, London.

Mayes, J.T. (1991) 'Cognitive Tools: A Suitable Case for Learning', In: Kommers, P.A.M., Jonassen, D.H. & Mayes, J.T. (eds), *Cognitive Tools for Learning*, Springer-Verlag, Berlin.

McAleese, R. (1990) 'Concepts as hypertext nodes: The ability to learn while navigating through hypertext nets', In: Jonassen, D.H. & Mandl, H. (eds), *Designing hypermedia for learning*, FRG: Springer-Verlag, Heidelberg.

McLaughlin, S. & Gero, J.S. (1987) 'Acquiring expert knowledge from characterised designs', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1 (2), pp. 73 - 87.

Merrill, M. D. (1983) 'Component display theory', In: Reigeluth, C.M. (ed). *Instructional design theories and models: An overview of their current status*, Lawrence Erlbaum, Hillsdale, N.J., pp. 279 - 333.

Merril, M.D., Kowalis, T., & Wilson, B.G. (1981) 'Instructional design in transition', In: Farley, F. & Gordon, N. (eds), *Psychology and education: The state of the Union*, McCutcheon, Chicago.

Midoro, V., Chiocciello, G., Olimpo, D., Persico, L. & Sarti, M. Tavella. (1988) 'Interactive video and artificial intelligence: A convenient marriage', *Programmed Learning and Educational Technology*, 25 (4), pp. 299 - 305.

Minsky, M. (1975) 'A framework for representing knowledge', In: Winston, P. (ed), *The psychology of computer vision*, McGraw Hill, New York.

Mittal, S. & Dym, C.L. (1985) 'Knowledge acquisition from multiple experts', *A.I. Magazine*, 5 (2), pp. 32 - 36.

Mokros, J. & Tinker, R. (1987) 'The impact of microcomputer-based labs on children's ability to interpret graphs', *Journal of Research in Science Teaching*, 24(4), pp. 369 - 383.

Mory, E.H. (1992) 'The use of informational feedback in instruction: Implications for future research', *Educational Technology Research and Development*, 40 (3), pp. 5 - 20.

Neale, I.M. (1988) 'First generation expert systems: a review of knowledge', *Knowledge Engineering Review*, 3 (2), pp. 105 - 145.

Nelson, T. (1974) *Dream machines / Computer Lib: new freedoms through computer screens - a minority report*, Theodore Nelson.

Nelson, T. (1987) *Computer Lib / Dream Machines*, Tempus Books of Microsoft Press, Redmond, Washington.

Newell, A. & Simon, H.A. (1972) *Human problem solving*, Prentice Hall, Englewood Cliffs, NJ.

Nielsen, J. (1995) *Multimedia and Hypertext: The Internet and Beyond*, Academic Press, Inc., Boston, USA.

Nisbett, R.E. & Ross, L. (1980) *Human inference: Strategies and shortcomings of social judgement*, Prentice-Hall, Englewood Cliffs, NJ.

Norman, D.A., Genter, S. & Stevens, A.L. (1976) 'Comments on learning schemata and memory representation', In: Klahr, D. (Ed.), *Cognition and instruction*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Palincsar, A.S. & Brown, A.L. (1984) 'Reciprocal teaching of comprehension-fostering and monitoring activities', *Cognition and Instruction*, 1(2), pp. 117 - 175.

Papert, S. (1980) *Mindstorms children, computers, and powerful ideas*, Basic Books, New York.

Parsons, R. (1994) 'Comparing and Contrasting the Development of Two Multimedia Learning Environments' Unpublished Project report, University of Plymouth.

Pask, G. (1975) *Conversation, cognition and learning: a cybernetic theory and methodology*, Elsevier, Amsterdam.

Payne, S.J. & Squibb, H.R. (1990) 'Algebra mal-rules and cognitive accounts for error', *Cognitive Science*, 14, pp. 445 - 481.

Peltu (1996) Article in an August edition of *Computing*.

Piaget, J. (1970) 'Piaget's theory', In: Mussen, P.H. (ed) *Carmichael's Manual of Child Psychology*, Wiley, New York.

Prawat, R.S. (1991) 'The value of ideas: The immersion approach to the development of thinking', *Educational Researcher*, 20(2), pp. 3-10.

Preece, J. (1993) 'Hypermedia, multimedia and human factors', In: Latchem, C., Williamson, J. & Henderson-Lancett, L., (eds), *Interactive Multimedia: Practice and Promise*, Kogan Page, London, pp. 135 - 150.

Rada, R. (1990) 'Expertext: Intelligent hypertext', *Intelligent Tutoring Media*, 1, (2), pp. 55 - 64.

Rada, R. (1991) *Hypertext: from Text to Expertext*, McGraw-Hill, London.

Regoczei, S. & Plantinga, E.P.O. (1987) 'Creating the domain of discourse: ontology and inventory', *International Journal of Man -Machine Studies*, 27, pp. 235 - 250.

Reigeluth, C.M. (ed) (1983) *Instructional-design theories and models: An overview of their current status*, Lawrence Erlbaum, Hillsdale, NJ.

Reigeluth, C.M. (ed) (1987) *Instructional theories in action: Lessons illustrating selected theories and models*, Lawrence Erlbaum, Hillsdale, NJ.

Reigeluth, C.M. & Stein, R. (1983) 'Elaboration theory', In: Reigeluth, C.M. (ed), *Instructional design theories and models: An overview of their current status*, Lawrence Erlbaum, Hillsdale, N.J., pp. 335 - 381.

Reiser, R.A. (1987) 'Instructional technology: A history', In: Gagné, R.M. (ed), *Instructional technology; Foundations*, Lawrence Erlbaum, Hillsdale, NJ.

Resnick, L.B. (1981) 'Instructional psychology', *Annual review of Psychology*, 40, pp. 631 - 666.

Resnick, L.B. (1989) 'Learning in school and out', *Educational Researcher*, February, pp. 13 - 20.

Rich, E. (1979) 'User modelling via stereotypes', *Cognitive Science*, 3, pp. 329 - 354.

Ridgeway, J. (1991) 'Of Course ICAI is Impossible..., Worse, Though, it Might be Seditious', In: Boyd-Barret, O. & Scanlon, E., (eds), *Computers and Learning* (An Open University Reader), Addison-Wesley Publishing Company, Wokingham, England, pp. 123 - 146.

Rieber, L.P. (1990) 'Animation in computer-based instruction', *Educational Technology Research and Development*, **38** (1), pp. 77 - 86.

Roblyer, M.D. (1988) 'Fundamental Problems and Principles of Designing Effective Courseware', In: Jonassen, D. H. (ed), *Instructional Designs for Microcomputer Courseware*, Lawrence Erlbaum Associates, London.

Rosch, E. (1978) 'Principles of categorization', In: Rosch, E. & Lloyds, B.B. (eds), *Cognition and categorization*, Erlbaum, Hillsdale, NJ, pp. 27 - 48.

Ross, P., Jones, J. & Millington, P. (1987) 'User modelling in intelligent teaching and tutoring', In: Lewis, R. & Tagg, E.D. (eds), *Trends in Computer Assisted Instruction*, Blackwell, London, pp. 32 - 44.

Roycroft, A.E. & Loucopoulos, P. (1985) 'ACCI - an expert system for the apportionment of close companies' income', In: Bramer, M.A. (ed), *Research and Development in Expert Systems*, pp. 127 - 139.

Royce, W.W., (1970) 'Managing the development of large software systems', In: *Proc. WESTCON*, San Francisco CA.

Rumelhart, D.E., & Ortony, A. (1977) 'The representation of knowledge in memory', In: Anderson, R.C., Spiro, R.J. & Montague W.E. (eds), *Schooling and the acquisition of knowledge*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Scardamalia, M. & Bereiter, C. (1985) 'Fostering the development of self-regulation in children's knowledge processing', In: Shipman, S.F., Segal, J.W. & Glaser, R. (eds), *Thinking and learning skills: Research and open questions*, Lawrence Erlbaum, Hillsdale, NJ., pp. 563 - 577.

Scardamalia, M., Bereiter, C. & Steinbach, R. (1984) 'Teachability of reflective processes in written composition', *Cognitive Science*, 8, pp. 173 - 190.

Schoenfeld, A.H. (1983) 'Problem solving in the mathematics curriculum: A report, recommendations and an annotated bibliography', *The Mathematical Association of America*, MAA Notes, No. 1.

Schoenfeld, A.H. (1985) *Mathematical problem solving*, Academic press, New York.

Seels, B., & Glasgow, Z. (1990) *Exercises in instructional design*, Merril Publishing Company, Columbus, OH.

Self, J.A. (1974) 'Student models in computer-aided instruction', *International Journal of Man-Machine Studies*, 6, pp. 261 - 276.

Self, J.A. (1987) 'The application of machine learning to student modelling', In: Lawler, R. & Yazdani, M. (eds), *Artificial Intelligence and Education, 1: Learning Environments and Tutoring Systems*, Ablex, Norwood, pp. 267 - 280.

Self, J.A. (1988) 'Student models: what use are they?', In: Ercoli, P. & Lewis, R. (eds), *Artificial Intelligence Tools in Education*, North Holland, Amsterdam, pp. 73 - 86.

Shavelson, R.J. (1972) 'Some aspects of the correspondence between content structure and cognitive structure in physics instruction', *Journal of Educational Psychology*, 63, pp. 225 - 234.

Simon, H.A. (1969) *The Sciences of the Artificial*, 1st Edition, MIT Press, Cambridge, Mass.

Skinner, B. F. (1938) *The Behaviour of Organisms: an Experimental Analysis*, Appleton-Century-Crofts, New York.

Skinner, B. F. (1968) *The Technology of Teaching*, Appleton-Century-Crofts, New York.

Sleeman, D.H. (1985) 'UMFE: a user modelling front-end subsystem', *International Journal of Man-Machine Studies*, **23**, pp. 71 - 88.

Sleeman, D., Kelly, A.E., Martinak, R., Ward, R.D. & Moore, J.L. (1989) 'Studies of diagnosis and remediation with high-school algebra students', *Cognitive Science*, **13**, pp. 551 - 568.

Smith, H. (1993) *The Use of Computerized Databases in Critical Inquiry by Pupils Aged 8 to 11 Years*, D. Phil. thesis, University of Sussex.

Smith, P.L. & Wedman, J.F. (1988) 'Read-think-aloud protocols: A new data source for formative evaluation', *Performance Improvement Quarterly*, **1**(2), pp. 13 - 22.

Sommerville, I. (1992) *Software Engineering*, Addison-Wesley, Wokingham, England.

Somekh, B. (1994) 'Designing software to maximise learning: what can we learn from the literature?', *Paper presented at the conference of the Association for Learning Technology, University of Hull, September 1994*.

Soper, P.J. and Bench-Capon, T.J.M. (1992) 'Using Hypertext to Interface to Legal Knowledge Based Systems', *Proceedings of the DEXA 92*, Valencia, Springer-Verlag.

Spector, J.M., Muraida, D.J., & Marlino, M.R. (1992) 'Cognitively based models of courseware development', *Educational Technology Research and Development*, **40**(2), pp. 45 - 54.

Spradley, J.P. (1979) *The ethnographic interview*, Holt, Rinehart and Winston, New York.

Thorndike, E.L. (1922) *The psychology of arithmetic*, Macmillan, New York.

Thro, M.P. (1974, March) 'Individual differences among college students in cognitive structure and physics performance', *Paper presented at the annual meeting of the American Educational Research Association, Toronto, Ontario, Canada*.

Tobias, S. (1985) 'Computer assisted instruction', In: Wang, M.C. & Waldberg, H.J. (eds), *Adapting Instruction to Individual Differences*, McCutchan, Berkeley, CA, pp. 139 - 189.

Trigg, R.H., Suchman, L.A., and Halasz, F.G. (1986) 'Supporting collaboration in NoteCards', *Proc. 1st Conf. Computer-Supported Cooperative Work*, Austin, TX, 3 - 5 December, pp. 153 - 162.

Tripp, S.D. & Bichelmeyer, B. (1990) 'Rapid prototyping: An alternative instructional design strategy', *Educational Technology Research and Development*, **38** (1), pp. 31 - 44.

Tversky, A. & Kahneman, D. (1974) 'Judgement under uncertainty: Heuristics and biases', *Science*, **185**(4157), pp. 1124 - 1131.

Twelker, P.A., et al (1972) 'The systematic development of instruction: An overview and basic guide to the literature. ERIC Document Reproduction Service, No. ED 059 629

Ullman, D.G., Dietterich, T.G. & Stauffer, L.A. (1988) 'A model of the mechanical design process based on empirical data', *AI EDAM*, **2** (1), pp. 33 - 52.

U.S. Bureau of Reclamation (1987) *Design of small dams (3rd ed.)*, United States Department of the Interior, United States Government Printing Office, Denver, Colorado.

Vassileva, J. (1991) 'An architecture and methodology for creating a domain independent, plan-based intelligent tutoring system', *Education and Training Technology International*, **27** (4).

Vygotsky, L.S. (1962) *Thought and Language*, The MIT Press, Cambridge, Mass.

Wachsmuth, I. (1988) 'Modelling the knowledge base of mathematical learners: situation-specific and situation non-specific knowledge', In: Mandl, H. & Lesgold, A. (eds), *Learning Issues for Intelligent Tutoring Systems*. Springer-Verlag, London, pp. 63 - 79.

- Wagner, W. & Gagné, R.M. (1988) 'Designing Computer-Aided Instruction', In: Jonassen (ed), *Instructional Designs for Microcomputer Courseware*. Lawrence Erlbaum Associates, London.
- Waterworth, J.A. (1992) *Multimedia interaction with computers: human factors issues*, Ellis Horwood, London.
- Watson, D. (1988) 'Development studies module', in Watson, D. (ed.), *Learning Geography with Computers*, Warwick, MESU.
- Welbank, M. (1983) *A review of knowledge acquisition techniques for expert systems*, Martlesham Consultancy Services, Ipswich.
- Wenger, E. (1987) *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann, Los Altos, CA.
- Whalley, P. (1990) 'Models of hypertext structure and learning', In: Jonassen, D.H. & Mandl, H. (eds), *Designing hypermedia for learning*, FRG: Springer-Verlag, Heidelberg.
- White, B. & Horwitz, P. (1991) 'Computer Microworlds and Conceptual Change: A New Approach to Science Education', In: Boyd-Barrett, O. & Scanlon, E. (eds), *Computers and Learning*, Addison-Wesley Publishing Company, Wokingham, England.
- Wilson, B. & Cole, P. (1991) 'A review of cognitive teaching models', *Educational Technology Research & Development*, 39 (4), pp. 47 - 64.
- Winn, W., Li, Tian-Zhu. & Schill, D. (1991) 'Diagrams as aids to problem solving: Their role in facilitating search and computation', *Educational Technology Research and Development*, 39 (1), pp. 17 - 29.
- Yang, Chia-Shing., Moore, D.M., & Burton, J.K. (1995) 'Managing Coursware Production: An Instructional Design Model with a Software Engineering Approach', *ETR&D*, 43(4), pp. 60 - 70.

Yankelovich, N., Hann. B.J., Meyrowitz, N.K. and Drucker, S.M. (1988) 'Intermedia: the concept and the construction of a seamless information environment', *Computer*, **21**(1), January.

Yankelovich, N., Meyrowitz, N. & van Dam, A. (1991) 'Reading and Writing the Electronic Book', In: Boyd-Barrett & Scanlon, E. (eds) *Computers and Learning*, Addison-Wesley Publishing Company, Wokingham, England, pp. 235 - 249.

Zahner, J. E., Reiser, R.A., Dick, W. & Gill, B. (1992) 'Evaluating instructional software: A simplified model', *Educational Technology Research and Development*, **40** (3), pp. 55 - 62.

Zissos, A.Y. & Witten, I.H. (1985) 'User modelling for a computer coach: A case study', *International Journal of Man-Machine Studies*, **23**, pp. 729 - 750.