COURSEWARE ON THE FLY

A thesis submitted to Trinity College,
University of Dublin,
For a Masters Degree in Computer Science.

Brian J Dillon,
Department of Computer Science,
Trinity College, Dublin.

September 1998.

DECLARATION

This thesis is entirely the work of the author, except where otherwise stated, and has not been submitted for a degree for a degree at any other University. This thesis may be copied and lent to others by the University of Dublin.

Brian J Dillon
September 1998

SUMMARY

Third level education is at the moment experiencing a period of rapid change. Innovative teaching methods, in which new technology will figure, are required to enable universities maintain a high standard of education in the face of increasing demands from students [TUC92]. The introduction of new technology, by exposing us to wider ranges of education possibilities should course reflection upon and possibly reconsideration of the manner and techniques employed. One key to effective computer aided teaching systems is the ability to dynamically adapt learning strategies to the needs of individual students. 'Courseware on-the-fly' seeks to provide 'just in time' course material composition based on student profiles while adhering to pedagogical rules.

1. ACKNOWLEDGEMENTS

I wish to thank the staff of the T.C.D. Computer Science department especially Mr Vincent Wade, my project supervisor, for all the assistance I received over the past year. I would also like to thank the postgraduate students who gave up their time to evaluate the system developed for ‘Courseware on-the-fly’, and my family for putting up with me.

CONTENTS

APPENDIX

2. INTRODUCTION

Third level education is at the moment experiencing a period of unprecedented change. The ever increasing numbers of applicants for third level courses is forcing education to move from the 'elite' system to a 'mass' system of education. With this increase in numbers and the changing face and demands of education, traditional methods of teaching will be unable to cope with the strain of maintaining a high standard of graduates [DEN96]. The strain placed on universities comes from a number of different areas, which include;

- The increasing cost in education, which has been detailed in practically every newspaper.
- The demand for more curricula in different areas. These is been driven by businesses who seek to recruit directly from college [DEN96].
- There is also a shortage of resources for Universities. This is caused by increases in the numbers of students but also in the increase in faculty members. This can been seen detailed in almost any student or campus newsletter.

For universities to continue to excel in their role of tutor and mentor they will need to introduce more innovative teaching methods making use of the new technology which is available. However, it is important to note that the application of technology is often misguided and can lead to unstructured teaching aids which are of little use to the student. Some of these 'misguided' teaching aids could if used inappropriately, serve to frustrate both student and teacher. An appropriate teaching aid should never serve as a 'showcase' for the new 'advanced' technology but should focus primarily on the pedagogics, using the technology as an effective enabler. New technology will succeed in enabling more effective teaching practice, only if the available opportunity to rethink the concepts of learning and teaching, which are being adopted, is not wasted. This rethinking should occur both separately from, to enhance general teaching practice, and in accordance with the technology available.

2.1 MOTIVATION

Information technology is so prevalent nowadays that it is hard for people in every walk of life not to come across it. With the ever decreasing cost of technology added to this prevalence people are becoming gradually aware of the social implications of information technology. Many people who are aware of computers and their benefits, need to become more proficient in their use. With this in mind educators are turning to computers so as to gain some of these benefits for teaching.

There are a number of factors which lead to the necessity of using computer technology within third level education. As previously stated third level education faces the problem of catering for increased demand from course applicants. There also exists the demand from those successful applicants for more beneficial and 'modern' courses. With these demands in mind third level institutions are finding that the traditional methods of education are no longer sufficient for their needs and the students needs. Lecturers are now required to provide more in-depth exposure of an extensive range of subjects to a wider audience within a limited time frame. This exposure must however be meaningful for the students and therefore the lecturers must also strive to constantly improve their teaching techniques.

With the movement from an 'elite' to 'mass' system of education more opportunities become available to people who have either left the educational system for some time, or to people who want to take their education onto a new level. Distance education and correspondence courses such as the 'Open University', which caters for thousands of students within the British isles, are becoming more and more popular. To cater for increased demands from students, such as the need for improved and new course material to keep inline with what industries require from new graduates, institutions such as the 'Open University' must consider technology aids. These institutions, as well as the more traditional third level institutions, must be careful not to let the standard of education drop with the increased demands from students seeking education. Unfortunately with distance education the student can feel isolated from the other students and the college. The lack of group participation on a course should be addressed carefully lest the pedagogical standard be inferior. This might be one place where technology does not have all the answers !

Another aspect of learning which facilitates the introduction of computer assistance is the need for industrial training. Billions of pounds are spent on training every year within both the skilled and semi-skilled sectors of industry. Employers require that their employees are up to date on all aspects of their business. Traditionally training and retraining meant leaving the office and going to classes. With the advent of information technology, training can now occur within the office which thus reduces the cost to the employer. With this new educational technology, employees can take a course whenever they have time or need to do so. Taking the course no longer involves taking time out to arrange schedules around courses, since the course can be fitted in around the employees schedule as required.

It is evident that information technology is required to facilitate distance education. Without the application of instructional technology on the Open Corpus the learning environment can be considered as;

- Being a 'vast' repository of information.
- Unstructured, with information being sometimes hard to find.
- Relying on experienced learners being capable of filtering unwanted information.

However information technology yields some of it's major benefits when used properly within the traditional third level education environment. With the increasing class sizes it becomes harder on lecturers to impart their knowledge to all the students. Computer assistance can provide additional help to students who require further clarification. With computer assistance students may find it easier to contact the lecturers, through e-mail, and therefore solve any problems that they may have. The application of technology in a Closed Corpus environment seeks to [IBR95];

- Emphasise the students interpretation more than that of the experts.
- Provide students with the opportunity to learn, rather than forcing then to try to learn.
- Provide easy update of course material.

- Provide students with help and guidance when required.

There have been many failed attempts to apply technology to educational needs. The reason for this has been the lack of regard for pedagogical considerations and the over emphasis on technology [TUC92]. The technology must be considered as a means to an end and not an end in itself. If the time is taken to devout adequate time to researching the pedagogical requirements as well as to finding the best technology to use, the best results for both learner and educator will be attained.

This project proposes to improve on previous attempts at using technology within education. One clear improvement is the support of dynamic courseware composition taking the students' own objectives into account.

2.2 OBJECTIVES

Instructional technology, by exposing the wider ranges of education possibilities, should promote reflection upon, and possibly reconsideration of, the manner and techniques of education being employed. The overall objective of this thesis is to investigate the application of 'Just In Time' (JIT) courseware composition based on student profiles and pedagogical rules. With this in mind 'Courseware on the fly' will be developed to show the possibilities of integrating technology and pedagogical techniques. This will involve;

1. The examination of current and previous educational and learning theory. The three main educational theories which will be considered are Constructivism, which sees learning as an adaptive process. Cognitivism which sees learners as an information processing organisms requiring programming and Behaviourism, which sees learners as devoid from any initiative of their own.
2. The consideration of various technologies some of which have been used previously for supporting education and others which were new at the time of development of 'Courseware on-the fly'.
3. Supporting and justifying any developments made with just in time courseware composition. This is the most important part of this thesis and will be considered in much more detail later.
4. Providing recommendations for future development in this ever growing area of computer aided learning.

2.3 DEVELOPMENT APPROACH

'Courseware on-the fly' can be divided into two separate streams which firstly must be considered separately and then together as one complete unit. These streams of thought are the 'Pedagogical Concerns' and the 'Technical approach'. The Pedagogic concerns involve issues such as the way in which students are being taught whereas the Technical approach focuses more on the logistics of this method of teaching.

2.3.1 PEDAGOGY

Examining the Pedagogic Concerns involves researching the various ideas and ideals in teaching practice. Previous failures in this area have been due to inadequate consideration of the pedagogic requirements of the system [NEW96]. The pedagogic concerns are more important than the medium supporting the transfer of knowledge. If the pedagogic concerns are not satisfied, then the system developed will be of little or no use to the student. It is no good to dump knowledge at a student without firstly considering the repercussions of this knowledge not been taught in a meaningful and helpful manner. To understand how to correctly impart knowledge from teacher to pupil the theories of learning must be considered.

In spite of the critical importance of Learning theory within education, the problem of explaining how learning takes place and analysing the factors which have influence on it, remains somewhat confused. Educators often unfairly blame psychologists for this confusion, because there are many conflicting explanations of learning each based on differing psychological theories. There are a number of reasons for this conflict. The first reason is that learning is a highly complex activity. The most explainable reason is probably that people have different experiences in life and therefore have developed their own methods of gaining knowledge. Most psychologists seem to agree that learning can be considered as ‘a relatively persistent change in an individuals potential behaviour due to experience’.

The psychology of learning can be divided into three separate theories; Behaviourism, Cognitivism and Constructivism.

The Behaviourist approach maintains that if psychology is to be an ‘exact’ science it must focus on the study of observable behaviour. This involves the study of the responses made by the individual and upon the conditions under which they occur. This approach seeks to find connections between stimulus and response or between response and reinforcement, (known as conditioning), therefore placing great emphasis on the environment.

The Cognitive approach claims that if learning is to be understood then study can not be confined to observable behaviour. This approach focuses on the learner’s ability to mentally reorganise his/her psychological field[1] in direct response to experience. This therefore emphasises not only the environment but the way in which the individual interprets and tries to make sense of the environment. The student in this sense is then an information processing organism which is actively and deliberately processing and categorising information being received.

Constructivism is the most recent theory to come to the forefront of studies involving education and knowledge acquisition. This view sees learning as an adaptive process. Constructivists view knowledge as being individually interpretable. Therefore when considering the make up of a course the student must be given freedom to learn as they want while still being given guidance which will enhance the learning activity.

2.3.2 TECHNICAL APPROACH

[1] I.E. The inner world of concepts and memories…

During the development of ‘Courseware on-the fly’ a number of technology challenges became clear. These challenges include;

- Storing the course material used in the dynamic courseware composition. It was decided that a Database storage system would be a huge improvement on the traditional ‘flat file’ based systems. This database adds flexibility and control to the system being developed. The design and organisation of this database will be discussed at a later time. The World Wide Web is the chosen medium upon which to present ‘Courseware on-the-fly’. The reason for this choice is that the Web has great flexibility and the ability to reach a wide range of audiences. With these two choices the system gains all the advantages of Web-enabled databases.

- The decomposition of this course material into fine grained units. The course material needed to be broken down so as to facilitate as many different combinations of the material as was required.

- One of the most important and difficult tasks facing Universities is shaping their courses to meet the individual requirements of students with varied experiences [WHI98]. One possible method of aiding this task is to maintain a model of each student. Student Profiling is a means of determining the current knowledge state of the student [VAN88]. In ‘Courseware on-the-fly’ the student model contains two types of information; What the students know and what they hope to achieve.

- The Pedagogical rules which were researched had to be incorporated into the various technologies employed. This is so that ‘Courseware on-the-fly’ proves to be a successful teaching aid.

The above challenges will be discussed in later chapters in much more detail.

2.4 ORGANISATION OF REPORT

This chapter has given a general introduction to ‘Courseware on-the fly’. It presented the primary objectives and motivations for this thesis. Below is a brief overview of the rest of this thesis and its structure.

Chapter 2.

This Chapter will give the necessary review of educational theory and practice. It will present and critique a number of well recognised learning theories.

Chapter 3.

This Chapter will describe in detail the technological choices which were available during the development of ‘Courseware on-the Fly’. This will involve an in dept look at JDBC and Netscape’s Live Connect.

Chapter 4.

This Chapter will give an example of designing and implementing a educational course. It will show how the various theories can be employed effectively.

Chapter 5.

This Chapter will give details of the Architecture design for ‘Courseware on-the fly’.

Chapter 6.

This is where the overall implementation of ‘Courseware on-the fly’ will be discussed. This involves examining how the courseware from Chapter 4 fits in with the Architecture from Chapter 5.

Chapter 7.

This chapter will be used to measure the success of ‘Courseware on-the fly’. It will describe the factors which are used in evaluating the success of the system.

Chapter 8,

This Chapter will give the overall conclusions drawn from this thesis. This involves revisiting the primary objectives and how they were met. The chapter will conclude with insights into how ‘Courseware on-the Fly’ could be improved as well as recommendations for future work.

3. PEDAGOGY

Before any work can be started on the technological aspect of 'Courseware on-the-fly' time must be spent examining the pedagogical considerations. The pedagogical considerations are probably more important than the medium used to support them. Many educational systems have failed in the past due to inadequate time spent examining the pedagogy and too much time developing the technical side of the system. This chapter will introduce and discuss the different views on education and learning, thus giving a good insight into how 'Courseware on-the-fly' came to be. The next section provides good insights into education in general. Education must be firstly considered as a whole and then broken down into it's various components and theories.

3.1 EDUCATION

The 'purpose and meaning' of education must firstly be considered. The purpose that we ascribe to education determines the methods that are used to aid students to acquire knowledge. One of the most important branches of philosophy is epistemology, (the theory of knowledge). Philosophers have made numerous attempts to discover the means by which we acquire knowledge. Socrates claimed that we cannot acquire knowledge through learning ! [POP72]

"One cannot learn what one does not already know since one already knows it. And one cannot learn what one does not already know, since if one doesn't know it, one cannot recognise it as truth when one learns it. Therefore learning it is impossible, and any knowledge that we can have we already know."

There are, according to Socrates, two main types of knowledge. The first is the 'Visible or Sensible' knowledge, (i.e acquired through the senses). This type is divided up into images, shadows and opinions. The lower level of this type is blurred and vague while the higher level is clearly organised. The problem is that none of this constitutes knowledge, because none of it is indubitable. This is because it is not understood in terms of the Forms or universals and all that can be reported is how it *seems* and what it *appears* to be. The second form of knowledge is 'Intelligible information' which deals with the Platonic ideas, where true knowledge is possible. The lower level here is the use of ideas as hypothesis without understanding their true nature. One assumes, but still lacks the knowledge of the nature of these universals. The highest level, complete knowledge, occurs when the idea is known in the sense of being fully aware of it and understanding its nature. Plato conceived that the universe is divided into 'appearance and reality', and our information is divided into 'opinion and knowledge'. "We can only have opinions about the world of appearance, but our souls can have true knowledge about the real world". Bishop George Berkley[2] agreed with these ideas by showing that our experiences are not independently existing material objects, but rather a series of ideas [POP72]. All we know about these ideas is what we perceive. It is therefore impossible to tell from what is seen, if they exist apart from the minds which perceive them. In fact Berkley pointed out that, 'We cannot conceive or imagine, in terms of experience, what it would be like for our sensations to exist apart from being thought of . Berkley's most famous phrase being, "esse est percipi'[3] , that is the existence of things consists in their being perceived [POP72]. Educators who have aspirations of effecting their students beneficially must remember to take these theories into account. Indeed if we are to believe Socrates all educators already 'know' these theories and just need to exercise their minds so as to become fully aware of the knowledge itself !

According to the Greek poet Pindar nothing is more difficult or important to us than 'to become man'. The chief task of education is therefore to shape the student, or to guide the evolving dynamism through which students shape themselves as human beings [MAR69]. Education is an art, and one which is very difficult to master. It does however by it's very nature belong to the sphere of ethics and practical wisdom. Education can therefore be considered as an ethical art. Jacque Maritain describes education as "a practical wisdom in which a determinate art is embodied" [MAR69]. Maritain considers the prime goal of teaching to be the conquest of internal and spiritual freedom to be achieved by the student. In other words the students liberation through knowledge and wisdom. Maritain views support those held much earlier by Aristotle who spoke of the independence which is granted to people by intellect and wisdom as being the perfection of the human being. It must, however, be noted at this point that the freedom of which Maritain writes is not a mere unfolding of potentialities without grasping anything of reality. Without aim or objective education is incomplete. It is therefore the teachers' task to impart their knowledge ensuring that the student is aware of the aim and objective of the material encompassed.

One of the common errors in education theory is the pragmatic[4] over emphasis in education. This pragmatic theory of knowledge cannot produce in the student anything but a scholarly scepticism equipped with the best techniques of mental training and the best scientific methods, which will get "used against the very grain of human intelligence", causing the student to distrust the idea of truth and wisdom. Dr. Hutchins, (1943), lists the "four cults", (Scepticism, Presentism, Scientism, Anti-intellectualism), as the offspring's of pragmatic domination over education [MAR69]. Modern pedagogy stresses the necessity of carefully analysing the human subject. The error in education begins when 'the object to be taught' and 'the primacy of the object' are forgotten. This results in "the cult of the means-not to an end, but without an end-only ends up in a psychological worship of the subject". Another error in education in general is that the college performs its educational task while "not infrequently making the student a victim of overwork or disintegrating chaotic specialisation often extinguishing the fire of natural gifts and defrauding the thirst of natural intelligence by dint of pseudo-knowledge" [MAR69]. The solution here is to make the college more aware and to acknowledge the necessity of mutual help and the inevitability of a reciprocal tension between the student and the college. College education has its own world, which essentially consists

[2] George Berkley (1685-1753) was a Fellow of Trinity College Dublin where he wrote his two most important works, '*A Treatise Concerning the Principles of Understanding*', and '*Three Dialogues between Hylas and Philonous*'.

[3] Latin - literal translation is "to be is to be perceived"

[4] Pragmatic - Dealing with matters with regard to their practical requirements or consequences.

of the dignity and achievements of knowledge and the intellect, that is, of the human being's root faculty. It is of this world of education that knowledge, which is wisdom, is the ultimate goal.

3.1.1 LEARNING

At this point it is necessary to consider learning, what it is and how it can be accomplished. As stated earlier Socrates claimed that knowledge could not be acquired by learning. However under this definition from Socrates, learning could be taken to mean the 'change' in a persons knowledge and understanding. Mayer agrees with this when he defines learning as "the relatively permanent change in a person's knowledge or behaviour due to experience" [STO71]. The study of learning has been an important focus since the early Greek philosophers. Pindar, said that there is nothing more important for us than to become ourselves. Thorndike claimed that, "Man's power to change himself, that is, to learn is perhaps the most impressive thing about him" [STO71]. The study of learning is of the utmost importance so that the requirements needed to help ourselves and others to become better students are understood.

Some views which are extremely inadequate and misleading propose that learning is 'what pupils experience through what teachers tell them'. This assumes a restriction of the term which is completely unjustified. This also makes the incorrect assumption that the dispensing of information by the teacher necessarily involves assimilation by the taught. Many psychologists accept that learning is all-pervasive and penetrates into most if not all spheres of human activity. Learning is an active process which, according to Driscoll, results in a "Persisting change in human performance or performance potential" and which occurs as a result of the person's interaction with their environment [NEW96]. There are three phases of learning which can be identified for the teacher to examine so as to adhere to some form of structured teaching. The first phase is the 'Planning Phase'. This phase focuses on what is needed to be learned along with when, why and how it is to be learned. The outcome sought here is a plan which should help to identify the gap between what the students know and what they need to know. The plan should also indicate how this gap could be bridged effectively. The second phase of learning is the 'Implementation Phase'. During this phase the plan from the previous phase is put into action. Depending on how well the first phase is done and the constraints introduced, this phase could be very straight forward or require a lot more thought and selection. The third and final phase of learning is the 'Evaluation Phase', where the effectiveness of the materials used and the overall learning accomplished is evaluated. Within this phase the plan and implementation are reviewed and suggestions for further improvements made. Unfortunately, learning is extremely difficult to measure because of the inherently different learners and the context and content of the material.

3.2 LEARNING THEORY

Learning theory is the set of principles explaining changes in human performance or performance potential in terms of their causes. There are three main theories to be considered which encompass most of the work which has been done on learning. These are Behaviourism, Cognitivism and Constructivism.

3.2.1 BEHAVIOURISM

Behaviourism began in the late 19th century. Watson claimed that, "the subject matter of human psychology is the behaviour or activities of the human being" [NEW96]. The changes in behaviour here are considered as the overt manifestations of learning or the 'evidence' of learning. The relatively permanent changes in people due to learning enable them to cope effectively with the regularities of the environment. If the regularities change then the patterns of behaviour will require changes to be made.

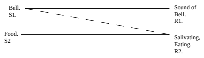
It would be incorrect to talk of learning and not consider Pavlov, a Russian physiologist who carried out the first and most important experiments which gave us insight into the learning process [STO71]. In 1880 when carrying out investigations into the digestive systems of dogs, Pavlov noticed that dogs salivated not only when the food was placed into their mouths but once they caught sight of food. This salivation at the sight of food involves the responding of the dog to a stimulus other than the one which should naturally evoke the response. Pavlov purported that the dogs had learned that the sight of food would lead them to be given something to eat. Later Pavlov modified the experiment by placing the dog in a harness in a room cut off from other stimuli. A neutral stimulus such as a ringing bell was presented before the dog was fed. Following repeated presentations, the dog soon 'learned' that the ringing of the bell would lead to it being fed and hence salivated upon hearing the bell. Pavlov called this a 'conditional response' since if the ringing of the bell did not immediately precede the dog being fed the response would die away. This conditional reflex is dependent on other factors for stability. The natural response, e.g. salivating due to the presence of food in the mouth, can now be called the unconditional response. Many others have resisted the idea that the behaviour of human beings can be explained, predicted or, worst of all controlled. Unfortunately, teachers traditionally have tried to do all three !

Around the same time as Pavlov, Thorndike was in America investigating learning in cats [STO71]. Thorndike placed a cat in a cage and placed food outside the cage. The only means the cat had to get out of the cage, and therefore get food, was a lever on the inside of the cage. In the beginning the cat exhibited what appeared to be random activities of scratching and banging on the cage. Eventually the cat came upon the lever and accidentally opened the cage. In subsequent trials the cat concentrated it's efforts on that side of the cage , where the lever was found earlier. Eventually the situation arose where once the cat was put into the cage it would go straight over to the lever and open the cage. This form of learning is known as 'Instrumental Conditioning'. Another American, C. L. Hull built upon the work of Thorndike and Pavlov. He argued that learning takes place through a process of 'need reduction'. The hungry cat, in Thorndike's cage, reduced its need for food by pulling the release on the cage. Hull's hypothesis links the activity of the brain with the behaviour of the animal. The animal's/person's needs, such as body temperature regulation, hunger and thirst are registered in the hypothalamus[5]. These needs are satisfied because of new 'connections'

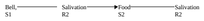
[5] The region of the brain which controls body temperature, thirst, hunger etc.

which are established in the cortex which leads to new patterns of behaviour, which end the state of need.

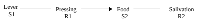
B.F. Skinner, who's work is most closely related to education, took up the study of reinforcement and in a series of experiments investigated the effect of the manipulation of reinforcement on various organisms. Thorndike, Hull and Skinner all approached learning in a similar way and their work fits into the broad category of what is called 'Instrumental conditioning'[6]. There is an inherent distinction between instrumental conditioning and classical conditioning as studied by Pavlov. [STO71]



S1 naturally evokes R1 which is insignificant.
S2 naturally evokes R2, but since S1 is presented at almost the same time as S2, S1 becomes linked with R2 so that S1 eventually elicits R2. Therefore;



Diag. (A). - Classical Conditioning.



Diag.(B), Instrumental Conditioning.

[S = Stimulus, R= Response, (S - R) = Reflex, → = 'is followed b

In instrumental conditioning the hungry animal, (e.g. a rat), is placed inside a 'Skinner box', which is a sophisticated adaptation of Thorndike's cage. The pressing of the lever will not bring about escape but will lead to food being dispensed. The lever acts as a stimulus which is completely neutral at first. The animal initially accidentally presses the lever. The pressing of the lever produces a reinforcer, (food), which then acts as the stimulus for the response of salivation. Since the animal is not sufficiently satisfied by the amount of food dispensed it continues it's investigative behaviour until it again operates the lever and produces no more food. It will continue to operate the lever until satisfied. At this point the teacher, Skinner, intervened and caused the reinforcer, i.e. the food to be presented in accordance with a variety of differing schedules. An example of this would include reinforcing every fifth response. Instrumental conditioning then differs from classical conditioning in that the required response is 'instrumental' in producing reinforcement. In the case of classical conditioning the reinforcement does not depend on the response made. The response may be considered as anticipatory rather than instrumental. Another distinction that could be made is that classical

[6] The term Instrumental Conditioning is sometimes referred as Operant Conditioning.

conditioning is a passive form of learning whereas instrumental conditioning is an active process. This distinction is drawn from the fact that the rat in Skinner's experiment obtained food because it 'pressed the lever'. Pavlov's dog is given food whether or not it responds to the stimulus. Skinner proposed that human behaviour can be explained in similar terms. If a parent wanted a child to repeat something a positive response in terms of praise would be given, or a negative response would follow the child misbehaving. Also with adults, successful accomplishment of a given task acts as a reinforcer of the activity which brought about the success. Thorndike said, 'Learning is connecting'. The learning investigations we have looked at so far have involved the animals making connections. If the mother of a new born baby feeds it when it cries, she will reinforce the crying so that every time it is hungry it will cry. There is no doubt that the principle of contiguity[7] is of utmost importance to education and learning. Learning cannot take place without some form of reinforcement. The complex mental activity associated with concept formation and problem solving depends on connections being made between the different aspects of phenomena and the responses constituting behavioural change. This results in the satisfactory formation of a concept and the solution of the problem.

Some people seem to dismiss the merits of behaviourism as inappropriate. Wittgenstein talked of 'Language on Holiday' when talking about behaviourism [TUC92]. His conception of the learner being a passive 'sponge' used to soak up the knowledge, i.e. The learner only responding stimulus and never initiating, is an inaccurate estimation of behaviourism. Skinner claimed that 'instruction refers to the arrangement of contingencies under which students learn'. Skinner, through his experiments with the rats, showed the practical advantage of reinforcement when teaching. Behaviourism should then neither be looked at with complete scepticism or as simply a means to simplify the approach taken to teaching.

3.2.2 COGNITIVISM

Cognitivism became a recognisable psychology around 1960. The reason for this recognition was because of the ever growing dissatisfaction with behaviourism. Karl Lashley when speaking at a symposium on 'Cerebral Mechanisms in Behaviour' argued that when people use language their behaviour is so rapid and continuous that it can not be controlled by external prompts alone [NEW96]. At this period in time, the computer was being developed and used as a powerful information-processing tool. The development of these machines influenced cognitivism greatly. Götz relates cognitivism closely to the "field of informatics" [TUC92]. Cognitivism sees the learner as an "information-processing organism", where the mind takes information in, stores it and then later does some sort of seek and retrieve operation.

Cognitivism in stark contrast to Behaviourism recognises the existence of internal mental phenomena. Learning according to the cognitive perspective can be described as a change in the knowledge, or perception, stored in a person's memory. Gardner and others made extensive studies of certain general methods of perceiving, the results of which are encompassed into the principles of 'cognitive control'. There are four main principles which are related to types of approach to complex perceptual tasks such as learning [VER71];

[7] Contiguity - being contiguous. The proximity of ideas or impressions in place or time, as a principle of association.

1. Field Articulation, which covers Witkin's 'field dependence' and 'field independence'. Witkin hypothesised that the passivity of some students is displayed in their 'field dependent' tendency to cling to the external framework environment of the visual field, whereas the self-reliance of other students, "the field independent students" is displayed in their ability to rely on their "own body sensations". Field articulation therefore relates to the selectiveness of attention. This means the capacity to direct attention actively and appropriately to the significant features of the field, disregarding irrelevant features. This is against the behaviourist view of passive acceptance of what is given.

2. Extensiveness of scanning. This is the tendency to deploy attention over a wide field as against narrow blinkered concentration. This is related to Piaget's 'centration'. Piaget made extensive studies into the errors made by students, adults and children, in judging visual illusions. By recording the eye movements of children he demonstrated that some students did not scan the pictures systematically. This, he proposed, was due to their gaze being "directed irregularly over the field", or it was 'centrated' upon a particular part of the field. It is claimed that wide scanning relates to the ability of humans to isolate knowledge and ideas from emotional connotations thus preserving it's accuracy.

3. Levelling and Sharpening. The former relates to perceiving things as they actually are and the latter to the tendency to assimilate percepts to the memory traces of previous percepts. An example of this is the use of analogies.

4. Tolerance of unrealistic experience. This refers to the control humans exercise in viewing something as they would normally when presented with it through distorted lenses.

At this point it is necessary to identify the two essential characteristics of human memory which cognitivists considered. The first being that knowledge is stored in an organised rather than random fashion. This can be recognised by simple tests such as getting the subject to write down names of people. The subjects will, most likely, start with their own names and then continue with their friend's names etc. The second characteristic is that memory is active rather than passive. This is completely contradictory to the behaviourists ideas of the mind as an empty bucket where information is simply poured in. Gredler identified three processes of human memory [NEW96]. Firstly, Attention which is the process of selectively receiving information, (i.e Field Articulation). Secondly, Encoding which involves translating the information into something meaningful which can be remembered, (i.e. Levelling and Sharpening). The final process Retrieval involves identifying and recalling information. Gredler's three processes can be seen to parallel the operations of a simple information processing machine. Cognitive learning is assumed to occur when new information has been encoded or existing information enhanced.

The primary responsibility of the Cognitive teacher is to organise and arrange external conditions so as to facilitate the student to encode and retrieve information. To do this the teacher must firstly organise the new information, since unorganised information will only serve to hinder the student. Secondly, new information should be linked with information which the student is already familiar with. This refers to the idea that new information which links with previous studies is accepted by the students as being more meaningful and therefore can be learned more easily. Lastly the teacher must use the technology available, such as computers, to guide and support the student's attention and retention of information. There are a number of techniques which are available to the cognitive teacher which can be used in conjunction with computer technology such as;

- Focusing questions. This is used typically at the beginning of a lesson as it focuses the student's attention to particular aspects of the new information.
- Highlighting. This involves different techniques such as underlining and italics to draw a student's attention to certain aspects.
- Analogies. This associates something new to something already learned. They are usually used to make concrete abstract concepts or information.
- Mnemonics. Any practical device used to make information easier to remember.

Considerable caution must be exercised when accepting the relationship of perceptual tendencies to personal characteristics which are addressed by cognitivism. Cognitivism also presents a somewhat inadequate representation of the learner. The inadequacies of Cognitivism become clear when the third theory of learning, Constructivism, is considered.

3.2.3 CONSTRUCTIVISM

The constructivist view sees learning as an adaptive process in which, "the learner's conceptual schemes are progressively reconstructed so that they are in keeping with continually wider ranges of experiences and ideas". The ideas of constructivism have been around since the 1890s when Dewey argued that, "education must be conceived as continuing reconstruction of experience" which occurs through stimulation by the demands of different social situations [NEW96]. Constructivism is under-pinned by a number of different principles of cognitive theories of knowledge structure and organisation. These contribute to the idea that the mind constructs it's own differing interpretations of information so it can draw inferences from them. Developmental psychology involves the idea that an individual's information constructs vary according to their cognitive development. Ausbel claimed that new information could only acquire meaning once it was incorporated into existing cognitive structures [TUC92]. These cognitive structures are hierarchically organised with lower order concepts being subsumed by more super-ordinate ones, rendering the cognitive structure increasingly complex and differentiated. It is thus that Ausbel considers prior learning of paramount importance. If the appropriate cognitive structures are inappropriate then new information can not be subsumed in an organised fashion and will be totally meaningless. An example of this is learning by rote, which is the mere mechanical repetition of something without attention to the underlying meaning of the material.

Constructivists describe learning as the change in the meaning constructed from experience. This could be mistaken for the cognitivist view. The critical difference is that cognitivists define knowledge as an objective representation of experience with the mind being akin to a mirror. Contructivists view knowledge as an individual's interpretation. That is that knowledge cannot be divorced from the individual and different students may look at information with differing perspectives. More recent theories on constuctivism propose that knowledge is represented within semantic networks, (called schemata). Not only do these networks function as storage devices but they also represent the inherent knowledge skill base. This representation becomes increasingly complex as the students increase their expertise.

Glaser and Bassok elaborated on this when they said, "Beginners knowledge of a domain is spotty, consisting of isolated definitions and superficial understandings of central terms and concepts. As proficiency develops, these items become structured and are integrated with past organisation of knowledge so that these are retrieved from memory rapidly and in larger units" [TUC92]. This semantic network concept explains some of the common misconceptions of students. These misconceptions occur when students attempt to fit new information into existing schemes which are inappropriate thus deforming the information in some way so as to fit it in.

Constructive learners are presented as people who actively interpret what they see according to their individual beliefs. Constructive teaching refers, then, to providing students with a collaborative situation within which they have both the opportunity and means to gain understanding. The two central characteristics which are widely accepted about constructivist education are.

1. The construction of good problems which "stimulate the exploration and reflection necessary for knowledge construction".
2. Collaboration. This refers to useful interactions of students with each other and with the teacher. The benefits from a learning perspective which can result from combined effort on a problem can be extensive if collaborative group work is allowed.

The primary concern of the constructivist teacher is to maintain a collaborative problem-solving environment. Constructivism has clear implications in instructional design. The students should be recognised as individuals who are ultimately in control of, and hence responsible for, their own learning. Another point to note is that since Constructivists recognise learning as an active process there should be some form of interaction during the course of learning. This interaction should involve the student's active reflection on what has been learned. Clearly this form of learning places a lot of the onus of learning on the student, rather than the material being poured in as if filling an empty bucket, (like in the behaviourist perspective).

Salomon and Globerson developed the concept of 'mindfulness' to explain the student's learning process [TUC92]. Mindfulness is defined as "The volitional, metacognitively-guided employment of non-automatic, usually effort demanding processes", it is, "a mid-level construct that entails motivational attitudinal and cognitive factors". So although mindful students employ effort this effort must be used productively to enhance deep learning. Mindful students are capable of monitoring their progress and understanding, make connections between new and already understood knowledge and are able to selectively transfer effective skills to new situations and tasks. This mindfulness therefore involves the complex interaction of attitudes, motivation and cognition. Attitude defines the expected opinion of a student when approached with a particular object. This is closely correlated with the motivation of a student to learn. This motivational drive in turn affects the students cognitive approach to learning. Extrinsic motivation refers to the fear of failure which normally results in learning by rote, whereas intrinsic motivation, associated with an interest in what is to be learned correlates better with constructivictic learning and is more beneficial to the student. To make something intrinsically motivational the learner must be careful not to divorce the material from reality. Brown claimed constructive learning (often referred to as Cognitive apprenticeship) methods "try to encapsulate students into authentic practices through activity and social interaction in a way similar to that evident - and evidently successful - in craft" [TUC92]. Cognitive apprenticeship models employ ideas such as situated modelling, coaching and fading. This refers to the situation where the student is given an example model, being aided to perform their own solutions and eventually performing tasks without aid.

Constructivism then demands that the learner engages in a mindful process of interactive learning. It is the newest and probably most applicable theory. The other theories such as behaviourism and cognitivism should not, however, be dismissed out of hand. These other theories have excellent foundations and much of what we know about learning is founded in these theories. What must be done is to make a complementary selection of ideals from all the previously mentioned theories.

3.3 BLOOM'S TAXONOMY

The most important and detailed work on providing guidance on the preparation of learning objectives has come from committees which were under the direction of Bloom [BLO56], Krathwohl [KRA64] and their colleagues. They produced a set of general and specific categories, arranged in hierarchical order, which encompass all expected learning outcomes from a lecture or class room. These categories can be further classified into three domains, the cognitive, affective and psychomotor.

3.3.1 THE COGNITIVE DOMAIN

This is the area of learning concerned with intellectual outcome. The cognitive objectives deal with the understanding and the application of knowledge. Bloom classified six educational objectives [BLO56];

1. Knowledge: This concerns the ability of the students to give evidence that they can recall or recognise facts, terms or theories presented to them.
2. Comprehension: This involves the understanding of the meaning and the intended use of the knowledge learned.
3. Application: This is the student's ability to apply the previously learned knowledge and comprehension in new situations.
4. Analysis: This is concerned with the ability of the students to breakdown the material into its constituent elements and to see the respective elements relationships with each other. Analysis could also be considered as the students ability to reach a conclusion based on the ideas or facts presented.
5. Synthesis: This can be thought of as the ability to combine and assemble existing knowledge elements into new and meaningful concepts.
6. Evaluation: Knowledge evaluation is the ability to judge the value of the material using criteria which are both explicit and coherent.

With the above taxonomy teachers are capable of examining whether or not their expectations of the students are adequate. For example is too much emphasis being placed on memorisation of facts at the expense of the more intellectual skills and abilities of the student.

3.3.2 THE AFFECTIVE DOMAIN

The categories here not only concern themselves with lecture or classroom learning but also the attitudes and values the students derive from the institution. The affective domain should not be seen as something separate from the cognitive domain, therefore it is necessary to explain Krathwohl's classifications [KRA64];

1. Receiving: This is the students willingness to attend the course.
2. Responding: This is the students willingness to participate actively in their learning.
3. Valuing: This differs from the evaluation in the cognitive domain in that it involves the students attitudes and social and moral judgements rather than the application of specific principles.
4. Organisation: This is the students ability to bring separate values and elements together for the purpose of comparison and relation.
5. Characterisation: This is the ability of the student to take the organisational level a step further and construct a coherent value system.

3.3.3 THE PSYCHOMOTOR DOMAIN

The committees that met under Krathwohl and Bloom originally had intended to produce a taxonomy of educational objectives in this third domain. The work was never finished, however, but work by people such as Simpson have made various attempts to remedy the omission [FON93]. The Psychomotor domain deals with the motor skills such as those required in sport and in manipulative exercises such as handwriting. It will not be covered any further here as information technology cannot adequately support this domain.

3.4 CONCLUDING REMARKS

Learning, as outlined in this chapter, is the relatively persistent change in an individuals potential behaviour due to experience. The three main theories of education and learning have been explained in detail. 'Courseware on-the-fly' includes aspects from all three educational theories but is primarily based on the Constructivist view. The next chapter explains the technology choices which were made while designing 'Courseware on-the-fly'.

4. INSTRUCTIONAL TECHNOLOGY

Instructional technology can be defined as applying scientific knowledge about learning to the practical task of teaching. If software aids are to be introduced to students they must add to, and in no way hinder, the intellectual development of the students. In this chapter the technologies used in achieving this are discussed. This will involve discussing the Web, which is the medium used to provide 'Courseware on-the-fly'. The technology choices made in 'Courseware on-the-fly' requires discussion as well as some examples of educational technology in use already.

One approach which could be taken when integrating computers and learning techniques involves cognitive apprenticeship. Cognitive apprenticeship employs the notion of modelling, coaching and fading (all of which were mentioned in chapter two in the context of constructivism). Computer Supported Intentional Learning Environment (CSILE), as developed by Scardamlia exhibits elements of cognitive apprenticeship. "The system is a form of hypermedia that allows notes entered as text, drawings, graphs and timelines to be retrieved, linked, commented on, rated and so forth" [TUC92]. This idea is based on procedural facilitation which provides the students with support while they develop complex cognitive skills. The supports are subsequently removed enabling the student to perform the skills unaided. Co-operative learning should be encouraged because learners themselves construct the knowledge base. In this form of learning discussion with others can help students to see that theirs isn't the only perspective from which issues can be viewed.

Another approach which involves the use of 'mind-tools' aims to encourage mindfulness[8] through focusing the students' attention on the conceptual structure of the material being learned. Examples of these tools include 'Learning Tool' and 'SemNet' [TUC92]. Learning Tool was developed by Robert Kozma and John van Roekel and is an "Electronic notebook that allows a learner to enter ideas (expressed in either text or graphic form) and to organise, reorganise, interrelate and search these ideas". When first run the notebook is empty and the student is presented with workspaces and tools to facilitate learning. SemNet is similar but the semantic network is conceptualised as "a ball, with richly connected concepts being pulled to the centre of the sphere due to their conceptual gravity" [TUC92]. The theoretical foundations in this case are that it encourages reorganisation of knowledge and forces the relations between concepts to be explicit.

4.1 THE WORLD WIDE WEB

The first choice which was made was to choose the World Wide Web as the medium for displaying the course material. The Web is the most widely used hypertext based system in the world. The Web is the result of many years of work in developing collaborative systems for information sharing. It provides a standard protocol, Hypertext Transfer Protocol (HTTP), for requesting information stored on any connected system whether it is local or remote.

[8] Mindfulness is the tendency to act in a particular way and the knowledge of how to do so effectively.

4.1.1 Hypertext and Hypermedia

Hypertext can be understood to be an approach to information management where data can be stored in a structured network of nodes connected by links. These 'nodes' can consist of images, text and video among others. Hypermedia is Hypertext with multimedia giving the ability to produce large and complex inter-referenced bodies of information. Hypermedia suits 'Courseware on-the-fly' where each course unit, which is rich in images and text, needs to be interconnected with other units.

The Web is based on HTML which is a language used in describing how documents should be structured and interrelated. It allows authors to write documents which are linked to others using HTML tags.

4.1.2 The Web and Education

One of the major advantages of the Web over other media is in it's ability to reach a wider audience. The Web has become one of the most widely used types of media over the past few years. It would be practically impossible for a student to go through college without having come into contact with the Web for research or e-mail purposes. The Web can be used to distribute learning material and to integrate collaborative tools for interactive educational courses and systems such as 'Courseware on-the-fly'.

There are two types of environment within which the Web can be used. The first is the Open Corpus environment where the student has free reign over what he /she looks at. The student must traverse a vast repository of information where information is often unstructured and hard to find. The second type of environment is the Closed Corpus environment. This is where the student is only allowed to view the information which is necessary to further their education.

OPEN CORPUS ENVIRONMENT

The Web was originally designed as an Open Corpus environment which facilitated information sharing among universities and research institutions. Although the Web holds vast amounts of useful data, the data is often unstructured and hard to find. The open corpus, as stated earlier, offers the student complete control over the viewing of the material. It is a cognitive tool, which allows students to explore and construct meaning in a self directional and motivational way.

In the open corpus environment many students find it difficult to understand and use the information provided by the Web. In some situations a student can be faced with too much information or undesirable material. The open corpus environment relies on the experience of the students and their ability to filter unnecessary information. The open corpus is perhaps best suited for a student's or lecturer's research / reference while not being necessarily great for practical educational experience.

CLOSED CORPUS ENVIRONMENT

The closed corpus Web has the same characteristics as normal courseware which executes on stand alone machines while gaining the advantage of being able to distribute the course content. The closed corpus environment on the Web places more emphasis on the students interpretation.

One of the main advantages of the closed corpus Web environment is that students can access course material from remote locations consistently and reliably. This means that the student may access the course wherever and whenever it suits. With multimedia capabilities being seamlessly integrated into the Web it is easy and cost effective to develop a course which provides a range of sensory stimuli for students. Since the course content is available to students 'indefinitely' there is no need for course handouts. It was for these reasons and the reasons outlined in the next section on 'Course Databases', that is was decided to develop 'Courseware on-the-fly' using the Web in the closed corpus environment.

4.1.3 adaptive Hypermedia

Adaptive Hypermedia is the process of integrating intelligent tutoring systems and hypermedia systems, such as the Web. Intelligent tutoring systems allow the modification of course content or presentation due to student interaction, whereas Hypermedia system learning concentrates on students interests and goals.

With hypermedia systems such as the Web it is possible for the user to feel disoriented and frustrated due to the lack of direction provided. Since different people learn in different ways a fixed set of road-maps guiding users is not the best solution. A more dynamic road-map structure is required which adapts in accordance with different users and their requirements. Educational material which is to prove useful and helpful for students should be tailored as much as is possible, to each learners capabilities and past history. Adaptive hypermedia systems, such as 'Courseware on-the-fly', increases hypermedia functionality by the use of student personalisation. This requires the use of intelligent tutoring techniques, in conjunction with knowledge about the students, to provide intelligent support and guidance to the Hypermedia navigation. For an adaptive hypermedia system to be effective it must reflect some features of the user in a user model. This user model must in turn be applied to the system so as to effectively adapt it for the user.

4.2 EDUCATIONAL SURVEY

Computers can be used in the design, development, delivery and evaluation of teaching and instruction. If computers are used for teaching and learning they have the ability to [NEW96];

- Enhance the quality of instructional materials.
- Reduce the time required to produce and reproduce instructional material.
- Increase the overall effectiveness of instructional materials through enhanced presentations.
- Combine various forms of media, e.g. graphics and video, into a single integrated instructional presentation.
- Store and quickly access vast amounts of information data.

In this section a number of educational systems which illustrate the use of technology in education will be described.

4.2.1 Virtual Universities

A number of third level institutions offer degree programmes in part or in entirety on the Web in response to ever increasing demands on their resources from college applicants.

WORLD LECTURE HALL

The World Lecture Hall, (WLH) provides a WWW index of course notes provided by many different educational institutions [WLH]. It was the University of Texas which first started this endeavour to comprehensively provide listings of Web course material offered by third parties. This course material includes; assignments, lecture notes, examination questions and multimedia textbooks. It should be noted, however, that the WLH does not provide any instruction itself. It is in effect a Web educational library of course material.

WALDEN UNIVERSITY

Walden University in Minneapolis offers on-line courses to students unwilling or unable to use the university campus [STE96]. Through this distributed infrastructure Walden university students can study for their degrees at their own pace and from any location in the World. Walden university also has an information network which is based on the Internet and connects a community of researchers and learners. It provides facilities such as access to research work and contact with student services on-line.

CORPORATE UNIVERSITIES

There are several companies involved in offering courses to compete with larger more established third level institutions. The problem with these however is that they can often be more biased to a particular product vendor making it difficult for a student to distinguish between sales pitches and genuine course material.

4.2.2 Virtual courses

There are numerous course being offered on the Web which attempt to provide courses which integrate instructional method and presentation uniformly. Two such courses will be reviewed here;

DUBLIN CITY UNIVERSITY

Dublin City University (DCU) developed a course on databases which had previously been taught using traditional lectures [SME97]. The module ‘CA309: Databases’ is taught to over 100 students in the third year of a four year degree programme.

The contents of the database course was organised into segments. Each segment is presented as a RealAudio track of between 2 and 20 minutes, requiring that ever player must have a RealAudio player installed. The presentation of the course material can be synchronised to the audio track being played. A transcript of the Audio file is provided which can be searched and short (30 second) segments can be replayed if required. As the course is structured loosely as a hypermedia document a number of shortfalls become apparent. Students could easily get disoriented since the course structure is confusing, no tutorial support is provided and the learning goals are not clearly defined.

TRINITY COLLEGE DUBLIN

Web courseware was developed by Trinity college for the purpose of aiding the teaching of Relational database concepts and SQL on the BA(MOD) computer science degree course. It comprised of a set of interactive Web pages containing text, graphics, animation and tutorial questions [WAD98]. The course contains several types of information [WAD97];

- Course Curriculum.
- Course usage instructions.
- A database of course modules.
- Road-maps showing sort paths through the course to achieve specific goals.

From results of student evaluation forms and examinations it was found that the majority of students benefited from this course.

4.2.3 virtual learning environments

To facilitate efficient and effective delivery of course material, environments have been offered, which can be easily tailored to deliver educational courses. A number of Virtual learning environments exist in the marketplace. These include;

- Symposium; This was developed by Centra Software and provides a content neutral, web based architecture for delivery of online training and learning [CEN].
- Pebblesoft Learning Environment; This environment, developed by Pebblesoft, provides the ability to store curricula in multi-media databases, set up instruction for students on a whole or part time basis and to track students progress [PEB].
- The Virtual Classroom [VCL]; This was developed at the New Jersey Institute of Technology (NJIT) and is adapted from the Electronic Information Exchange Systems (EIES) with special software structures which support collaborative learning [HIL95].
- TopClass, a product developed by Web Based Training (WBT) systems, was selected as PC Week labs best web-based training solution and will be discussed here to illustrate the general approach taken.

TOPCLASS

TopClass is a commercially available software package which is designed to manage the delivery and support of training and education over the Internet and corporate or campus intranets using the Web [TCL98]. It also provides facilities to manages all students and instructors on the server as well as assigned courser material.

TopClass contains its own outlining environment which permits authors to construct courses entirely on-line from a standard Web browser. Courses are composed of Units of Learning Material (ULM). These ULMs can consist of pages, exercises and other ULMs. ULMs can be used to provide lecturers with the facility to alter a particular student's course without affecting the rest of the students in the class. ULMs can be exported to / imported from one course to another. TopClass navigation is facilitated through the hierarchical ordering of the ULMs. Other features provided by TopClass include;

- Student tracking and monitoring through the use of tests. These tests can be used to determine the direction the student takes within the course.
- Plug 'n' play capabilities allow the easy distribution and transfer of course material and full courses to other servers.
- Collaboration tools such as bulletin boards.
- Built-in e-mail to facilitate student communication.

The major drawback associated with using TopClass in a large third level institutions is that multiple servers would be required to cope with the diverse number of courses on offer, e.g. math, computer science, economics. Since there is no co-operation between different servers the is no facility for central integration and control of students. Another drawback of TopClass is it's inadequate aid for courseware development making it only a management system for educational courses.

4.3 COURSE DATABASES

Databases were chosen over traditional flat file based storage because they facilitate easier course reorganisation. Web-enabled databases allow more flexibility in changing the method by which courses are offered and organised.

WEB-ENABLED DATABASES

Web-enabled databases provide opportunities to change the way educational courses can be offered and managed. If the Web was not used as the medium for offering the courses, their appearance could differ depending on which operating system was used to display the course. Remember the Web was originally developed for the purpose of sharing research. It is the huge interest from students in the Web and the wide deployment and use of the Web for personal and business usage that are driving the use of the Web and Web-enabled databases in education. The major problem for the deployment of Web-enabled databases for education has been the lack of co-ordination among college faculty members, technology support personnel and administrative decision makers [ASH98]. Databases provide, for Web publishing, the perfect method for publishing content on-the fly. The database provides the flexibility to change;

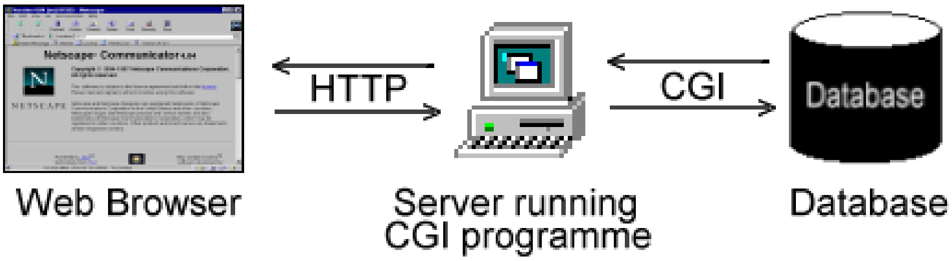
- The Content of each of the Units in a course.

- The Way in which these Units make up a course.

These can be changed without having to rewrite an entire course or course unit.

4.3.1 Database Connectivity

As explained in chapter one, the two choices for Web-enabled database connectivity are the Common Gateway Interface (CGI) and Java Database Connectivity (JDBC). CGI is a standard for external gateway programmes to interface with servers.

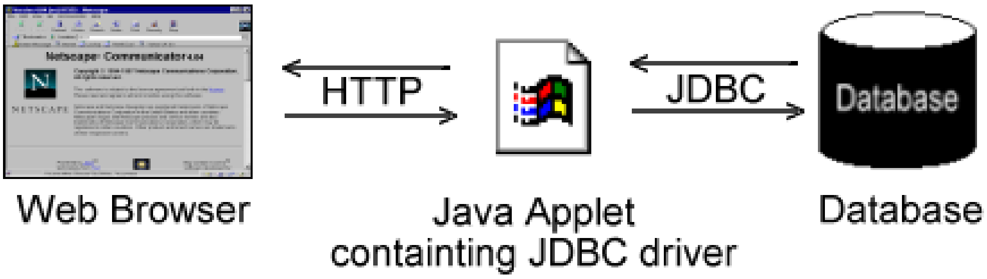


(Diagram 3.1, Using CGI)

The CGI approach is probably the best known approach for connecting a database to the Web. A browser requests that an executable to be started at the server side. This external application then performs the database access, taking the given input parameters and delivers the result back to the server. This method usually involves writing the software to access the database in C or Perl. There are two main problems associated with the CGI approach.

- The sever load can be heavy as each invocation of the CGI programme starts a new server process.
- Transaction management is inadequate.

A product which is called Fast CGI has been released to deal with some of these performance problems.



(Diagram 3.2, Using JDBC)

The second approach in which a database can be connected to the Web is to use JDBC, Java Database Connectivity. JDBC is the way in which Java can be used to interact with a database. The JDBC approach involves the Web browser downloading an applet. The applet can then perform the database access from the client side, relieving the server from the database access. This approach affords many advantages;

- Full transaction management and concurrency control can be offered.
- The platform independence of Java and the support by browsers ensure availability on heterogeneous platforms.
- JDBC is designed to allow access to multiple databases if the need arises.

The next section here will discuss JDBC and all it's possible implementations in detail.

4.4 CONNECTING WEB-DATABASES USING JAVA

Java Database Connectivity (JDBC) is a standard set of Java classes which provides vendor-independent access to relational data. The JDBC standard was developed by Javasoft and Oracle among others. JDBC provides an application programming interface (API) by which it is possible to communicate with databases. The JDBC API was designed as being a set of vendor-neutral Java interfaces to relational databases. JDBC vendors in turn provide different implementations of these interfaces which get classified as different JDBC driver types.

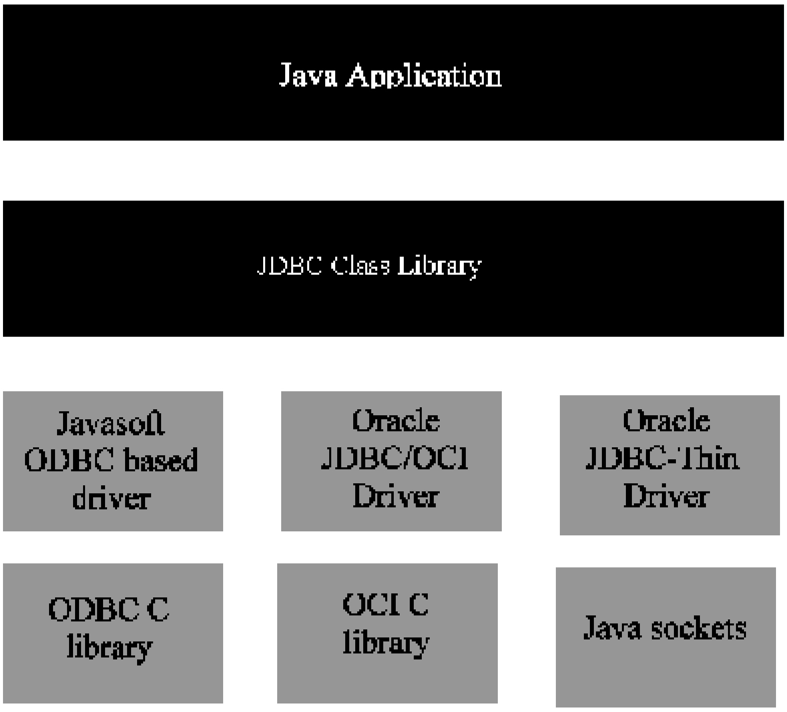
JDBC drivers are implemented in two layers. The 'lower' layer provides a Java interface, called DBAccess. This layer handles buffered communication, such as DB queries, with the database. This DBAccess API is quite small and only provides a few entry points. The 'upper' layer calls the DBAccess entry points and thus implements the JDBC interfaces. This layer is large and provides multiple entry points for fetching columns from queries among others. This upper layer implements most of Javasoft's specified JDBC logic such as data conversions between Java and SQL types, etc. Most implementations of JDBC drivers share the same upper layer. However, the lower layer which provides the actual communication with the database is where the different approaches distinguish themselves.

From a programmer's perspective, JDBC offers three important elements of functionality.

- Comprehensiveness: The JDBC classes provide standard features such as simultaneous connections to several databases, simple queries and calls to stored procedures.
- Application execution control: JDBC is a fine grained interface which provides control over the application's profile including it's memory and network usage allowing the performance and efficiency to be fine tuned.
- Interoperability: JDBC provides a database independent API standard that allows applications to be written irrespective of the underlying RDBMS. JDBC can then be considered as an interoperability solution in a multivendor database environment.

4.4.1 JDBC techniques

A number of different types of JDBC drivers exist which communicate with databases in several different ways. These drivers can be categorised into two categories. The first being those designed for client-server Java applications and Java-based middle-tiers. This category is then subdivisible into two driver types; JDBC-ODBC Bridges and native-API drivers. The second category, those designed for use with Java applets, have driver types such as the Net-protocol pure Java driver and the native-protocol pure Java driver. In this chapter three alternative implementations of JDBC will be examined for use in the 'Courseware on the fly' development. These implementations, illustrated below, include a JDBC-ODBC bridge, the Oracle JDBC/OCI driver and Oracle's Thin JDBC driver. Oracle drivers were chosen her since they are the most popular drivers available.



(Diagram 3.3)

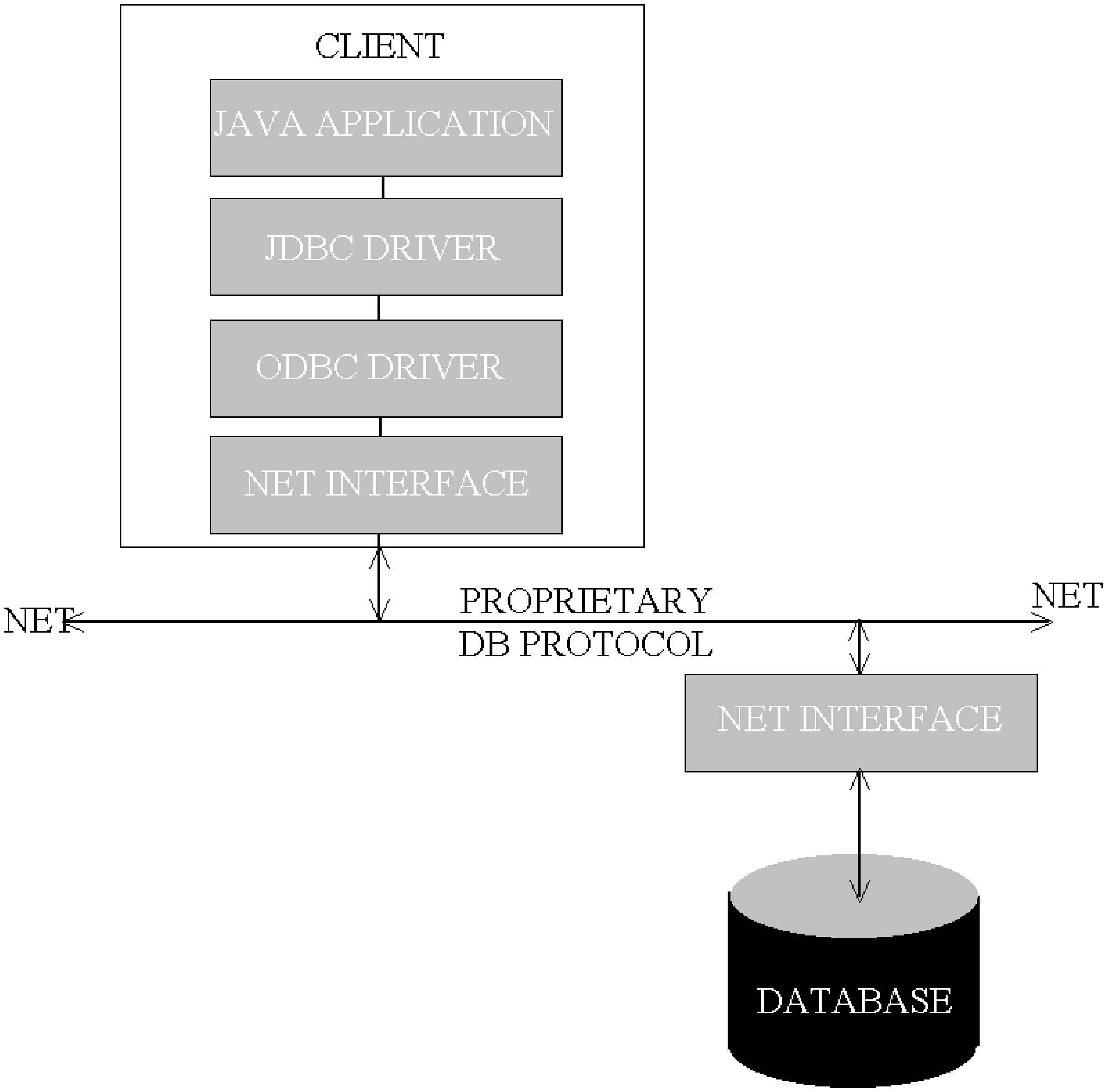
4.4.2 JDBC Drivers for Java Applications and Middle-Tiers

This first type of JDBC driver is used in Java development for traditional client-server application development in an Intranet environment where the Java client would be installed on a client machine such as a workstation. This first type of driver is also used in development of Java-based middle tiers such as Java application servers. Here a browser-based client communicates with a Java-based middle tier using a stateless protocol such as HTTP or a stateful protocol such as IIOP. The middle tier is responsible for executing the Java application logic and communication with the database server using JDBC.

The main way that this type of driver is implemented is using a JDBC-ODBC bridge which was developed by JavaSoft to take advantage of the large number of ODBC[9] enabled data

[9] This is Microsoft's way of connecting the Web to a database.

sources. Java applications are written using the JDBC API. These calls are then passed to the JDBC-ODBC bridge and converted into C language ODBC APIs. The ODBC calls are then passed onto the ODBC driver for database communication.



(Diagram 3.4, JDBC-ODBC Bridge)

The main advantage of using a JDBC-ODBC bridge is that with the JDBC calls being converted into ODBC calls, applications can easily access databases from multiple vendors by choosing the appropriate ODBC driver. There are however three primary limitations of JDBC-ODBC bridges:

- Manageability Limitations: The JDBC-ODBC bridge adds significantly greater complexity to managing and administering a client-side installation caused by the complexity of ODBC drivers which consist of a driver manager, the driver itself and associated utilities.

- Overhead: This is probably the most important limitation of the JDBC-ODBC bridge. Connecting to a database using this method involves considerable overhead and complexity since calls must go from JDBC to the bridge to the ODBC driver and finally from ODBC to the native client API to the database.
- Functionality: Lastly the JDBC-ODBC bridges limit themselves to the subset of functionality provided by all databases. By providing this maximum number of common functions, they prevent users from exploiting database specific advances such as performance and scalability improvements or object relational functionality.

4.4.3 using the call level interface

This is a type 2 JDBC driver that is an implementation of a two-tier driver that connects the client with the database using vendor supplied call level interface, (in this case the Oracle Call Interface, OCI). This type of driver is for client-server Java applications and Java-based middle-tiers.

The JDBC/OCI driver converts JDBC invocations to the OCI which is in turn sent over SQL*Net to the Oracle database server. SQL*Net is Oracle's way of communicating from computer-to-computer. It allows connection to a set of databases on a local machine or on any remote machine. It eliminates the boundaries in a computer network created by heterogeneous systems. SQL*Net allows two or more servers to communicate, so that a server may become a client to another server as well as allowing connections between distributed databases. All this is made possible by SQL*Net taking care of the network connections and communication protocols required in communication over a computer network. Through the use of SQL*Net, the JDBC/OCI Driver can achieve much higher throughput than competing JDBC solutions that interface to ODBC.

In order to make calls to the OCI the driver must use a layer of C, therefore, the JDBC/OCI driver transforms calls from Java to C. To make this possible the driver is written in a combination of Java and C which prevent it from being made downloadable. Another drawback of this type of driver is that it requires that the OCI libraries, SQL*Net and CORE libraries should be installed on the machine where the JDBC driver is installed. This implies installing them either on the client machines or the middle-tier Java application server. The JDBC/OCI driver is suitable for client-server Java applications and middle-tiers but not for applets because of this drawback.

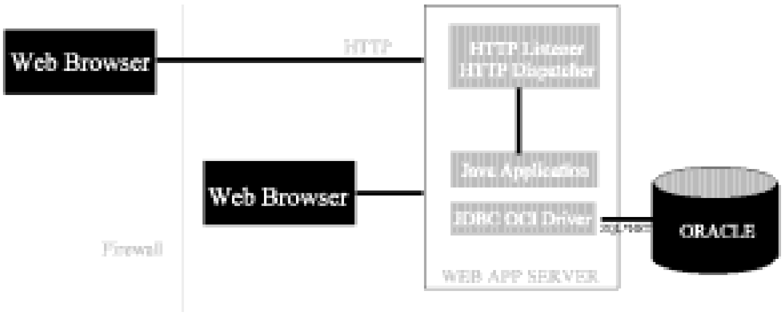
The benefits of using the JDBC/OCI do however far out-way the aforementioned drawbacks:

- Performance: The JDBC/OCI driver gains a number of the OCI's performance optimisations which are critical to delivering the high performance necessary for enterprise Java applications. These optimisations include reductions in the number of client to server round trips for binds, also allowing row prefetching and batched statement execution.
- Datatype Support: Being an Oracle product the JDBC/OCI driver provides support for Oracle-specific datatypes such as ROWIDs.
- Character sets: Following on from the last point, JDBC/OCI provides complete support for global character sets allowing Java applications to access databases that use any Oracle character set. This is achieved by converting the characters to Unicode 1.2.

- Firewall support: The JDBC/OCI driver works from within a firewall as well as outside the firewall. The firewall implementations which it works with are those which are SQL*Net certified, e.g. Raptor.
- SQL*Net Version support: Since it is targeted to support client-server Java applications the JDBC/OCI driver supports all SQL*Net adapters. It also supports all the features of Oracle's Advanced Networking Option including encrypted SQL*Net.

Since the purpose of this JDBC appraisal is to find the best JDBC type driver to work over a Web browser, it is necessary to take a look at the two possible 3-Tier configurations of the JDBC/OCI driver.

JDBC/OCI DRIVER 3-TIER CONFIG 1.



(Diagram 3.5, JDBC/OCI Config1)

The diagram above indicates how the JDBC/OCI driver can be used within a tree-tier configuration in an Intranet or Extranet[10] setting. The client, the Web Browser, communicates with the web server, (An Apache Web Server is used for 'Courseware on-the-fly'), using the HTTP stateless protocol. The Java application and the JDBC/OCI driver are installed on the web server. The application can be invoked in a number of ways:

- A CGI script could be used.
- If the application was deployed on Oracle's Web Application Server Oracle's Web Request Broker API could be used.
- Or an IIOP 'style' invocation.

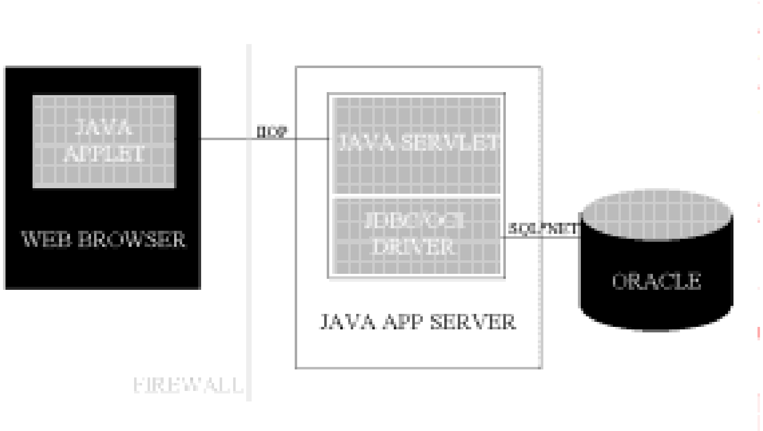
Once the Java application has been invoked, it communicates with the backend database server using SQL*Net. With this application deployment three important issues arise.

1. State Management: In this configuration, the communication between the client and middle-tier web server occurs via HTTP, while the communication between the Java application on the web server and the database occurs in a stateful manner. This causes the need for explicit management of all state information as well as handling of all the relationships between transaction state and various HTTP sessions.

[10] Extranet = Where the web server is located behind the firewall.

2. Response Time: Since HTTP is a stateless protocol, a simplistic Java application would essentially require a user to establish a database connection login each time. With HTTP's stateless nature this creates a severe performance problem, because of the round trip between the Web server and database server as well as the database login per HTTP request. This could be prohibitively expensive and needs to be avoided. The solution is to establish the database connection and login on the initial invocation. This login will be cached for future use with subsequent HTTP requests from the same user.
3. Scalability: Rather than establish a separate database connection for every web browser it is possible to pool the state associated with multiple clients on a single or a number of connections to the backend database. By doing this the database server memory usage is optimised and the number of clients, browsers, that the application can support is significantly increased.

JDBC/OCI CONFIG. 2



(Diagram3.6, JDBC/OCI Config 2)

The alternative three-tier architecture with the JDBC/OCI driver is shown in the diagram above. In this configuration the application logic is divided between Java applets and servlets. These applets and servlets typically communicate using either RMI or CORBA, (IIOP). The applet is invoked in the Web browser, (using the <applet> tag in HTML). This applet then communicates with the Java servlet that is deployed on the middle-tier web server in a stateful manner using, say, CORBA. The servlet in turn communicates with the Oracle database server using the JDBC/OCI driver.

As seen the primary difference between the two configurations of JDBC/OCI discussed is their relevant stateful/stateless nature of the communication between the browser and the Web server. The second configuration which involves using IIOP gains the advantage over the first configuration here.

4.4.4 java only drivers

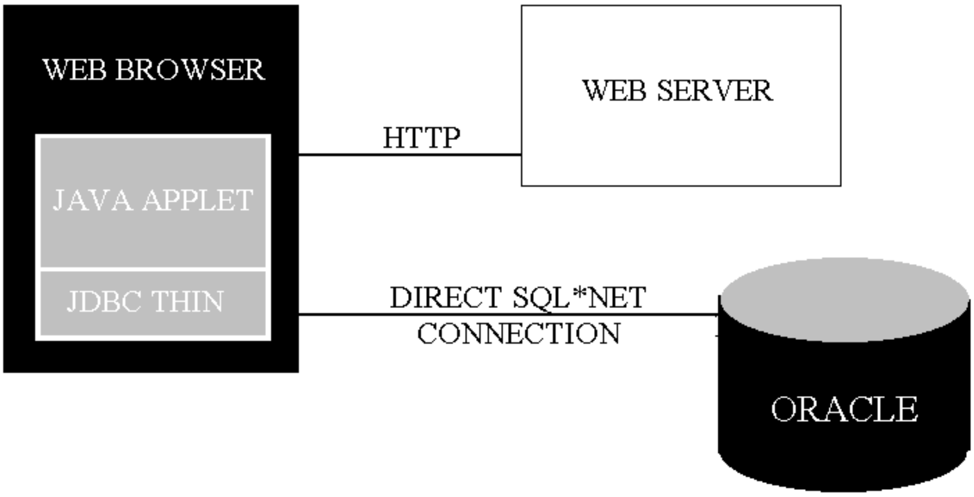
Here an Oracle type 4 JDBC driver which is for use with Java applets or applications will be considered. This JDBC driver is completely written in Java and is designed to be small so that is downloadable within a Java applet. The Uniform Resource Locator, URL, in the

Browser is selected. This URL would contain the <applet> tag. The Java applet and the JDBC-Thin driver are downloaded into the browser from the Web server. The JDBC-Thin driver then establishes a direct connection to the Oracle database server over Java sockets. It provides it's own lightweight implementation of a TCP/IP version of Oracle's SQL*Net/Net8 protocol. This results in the initial stateless HTTP request to download the applet and the JDBC driver from the Web server, being superseded by stateful communication between the browser and the database once the JDBC-thin driver is established.

The JDBC-thin driver like the JDBC/OCI driver has support for operation both inside and outside of firewalls as well as support for Oracle-specific datatypes and other global character sets. The JDBC-thin driver has two main beneficial features over the other two previously discussed drivers are;

- No library installation: The JDBC-thin driver is designed to be completely downloadable in applets and therefore unlike the JDBC/OCI driver it does not require pre-installation of support files such as SQL*Net or CORE on the client desktops.
- Performance : The JDBC-thin driver provides a number of performance optimisations which include reductions in the number of client to server round trips for binds, and describes row prefetching and batch statement execution. These optimisations prove necessary for high performance Java applications.

The JDBC-thin drive has restrictions on it's usage which the JDBC/OCI driver does not where the driver only works with TCP/IP based networks.



(Diagram 3.7, JDBC-thin)

The above diagram illustrates the configuration in which the Thin JDBC driver could be used for 'Courseware on-the-fly'. The Java applet and the JDBC-thin driver would be downloaded into the browser once the user initiates the start of the course. The JDBC-Thin driver communicates directly with the database server. The web sever and the database server can if required be on physically separate machines, (If the Java applets are signed for security reasons), or on the same machine. In an Intranet deployment the entire configuration is behind a firewall. In an Extranet deployment, however, only the web sever and the database are both behind the firewall.

This configuration raises a number of issues which need to be discussed at this point. The first issue is ‘State Management’. In the above configuration, the initial communication between the client/user and the middle-tier web server occurs via the stateless HTTP protocol. A direct stateful connection with the database is established, however, once the JDBC driver is downloaded thus eliminating the need to explicitly handle the relationship between transaction state and various HTTP sessions. The second issue is the scalability of this configuration. Since the JDBC-Thin driver establishes a separate connection from the web browser to the database server, there are two ways in which a user can support a large number of clients concurrently connecting to a single database. The first way being using the Net8 Connection manager which essentially multiplexes connections between a number of different clients to a single physical database connection. This reduces the number of direct database connections and therefore provides greater scalability. Secondly, Oracle8 database servers allow users to transparently leverage a feature called connection pooling. In this instance the Oracle Multi Threaded Server’s dispatcher establishes a pool of connections internally and hands connections from inactive to active clients as required thus supporting more clients.

The final issue concerning the JDBC-Thin driver configuration under discussion is Security. The communication between the applet using JDBC-Thin driver and the Oracle database is over TCP/IP sockets. For this connection to occur the web browser where the applet is executing must allow socket connections. In Netscape 3.0, which supports JDK 1.0.2, an applet can only open sockets to the host from which is was downloaded. To avoid security violations the web server and the Oracle server must be on the same machine. With Netscape 4.0, which supports JDK 1.1.1, this restriction is eliminated with the introduction of signed applets, enabling deployment of the web server and the Oracle database on separate machines. The applet can request socket connection privileges and if the user grants them the applet can connect to the database running on a different host from the web server host.

It is now necessary to explain some of the benefits which Oracle’s JDBC drivers provide over other JDBC drivers when used with an Oracle database. These benefits include;

- Both the Oracle JDBC/OCI and JDBC-Thin drivers provide all the JDBC entry points specified in the JDBC standard unlike other JDBC drivers which often do not support streaming long columns. Furthermore the drivers are JDK 1.0.2 and JDK 1.1 compliant.
- The JDBC-thin driver has been certified with Netscape Navigator and Internet Explorer as well as being certified with Java VMs such as JDK. These certificates mean that the Oracle’s JDBC driver can be used by the vast majority of clients which wish to gain access to the database.
- Being an Oracle standard means that they provide access to both the standard SQL types as well as Oracle-specific types.
- Both drivers can be used to access a wide variety of heterogeneous data sources. With the use of Oracle’s Transparent Gateway products both drivers can provide connectivity to in excess of 25 different enterprise and legacy databases such as Ingres and CICS.

The most important benefits from using the Oracle JDBC drivers come under the heading of performance. Anyone that has worked with Java applications will understand that performance is one of the most critical deciding factors when discussing Java compliant tools. There are three main ways in which the Oracle JDBC drivers are optimised with performance in mind. The first is ‘Direct Database Access’. Unlike the JDBC-ODBC bridge drivers which require multiple layers to access the databases, Oracle’s JDBC drivers use SQL*Net thus improving access time considerably. Secondly a number of extensions to JDBC are provided to improve performance. These include;

- Bulk Operations: This involves row pre-fetching.
- Batch Execution: Users are allowed batch INSERTS and UPDATES.
- Query Column Definitions: This is where the user is allowed to inform the driver of the types of columns in an upcoming query, thus saving a separate definition round trip to the database.

The third and probably the most important way that performance is optimised is with the size of the JDBC-Thin driver. The JDBC-Thin driver has been tightly controlled to be approximately 300K uncompressed. This helps to reduce the time required to download the applet containing the JDBC Driver. In an Extranet environment the streaming of the driver through the firewall can impose a severe performance penalty. The size factor helps to improve the speed of Java execution in the Java VMs integrated with browsers.

4.5 CONNECTING JDBC TO THE USER

Before Java Script, a problem existed where people were able to use HTML, Hypertext Mark-up Language, which could be processed by web browsers. HTML was powerful but was not on the same level as programming languages were. To solve this Netscape introduced a product called Live Script. This ‘Live Script’ enabled HTML to become more powerful and stronger than was expected. Live Script then became known as Java Script. Java Script is basically a programming language, that can understand HTML, which breathes life and intelligence into web pages. Whenever the web browser encounters Java Script instructions it simply calls the Java Script interpreter. This interpreter then can decipher the Java Script code.

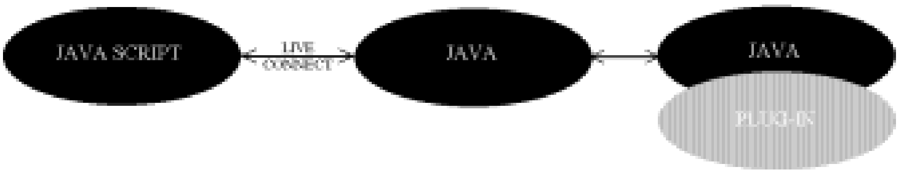
The question which arose then was; how can Java Script call the original Java code ? (i.e. the classes that come along with Java, not the Java programmes that are created by the developer). The answer to this question was simple. Since the original Java classes are built into browsers such as Netscape, they can be merged with the built in Java Script classes and called as one. The big problem, however, which arose was how to call the Java classes written by the Java developer. This problem is compounded by the fact that Java understands only Objects. To solve this Netscape introduced a new product called ‘Live Connect’.

4.5.1 Live connect

Live Connect is basically the means by which Java Script and Java work together for their common good. With the use of Live Connect a number of things become possible:

- Java Script programmes can interact with the standard Java system classes built into the browser without the need for merging the Java and Java Script classes as discussed earlier.
- Java Script programmes can interact with Java applets. This is the main reason for choosing it for the development of 'Courseware on the Fly'. Java Script can invoke public methods of the applet which can then be used to access the Database using JDBC technology as discussed earlier.
- Java Script can interact with Java-enabled plug-ins.
- Applets and Java-enabled plug-ins can interact with Java Scripts. This allows the possibility for Applets to write back the HTML pages on-the-fly.

Live Connect handles all the required communication and data type conversion that takes place in order to allow Java and Java Script to work together. It is a communication framework glue which opens up many possibilities for communication among Java, Java Script, and Java-enabled plugins. It should also be noted here that Live Connect does not allow direct communication between C/C++ plugins and Java Script. That is why the plugins must communicate with Java Script through the Java programming language. The diagramme below illustrates how Live Connect can be used in all of the afore mentioned situations.



(Diagram 3.8, Live Connect)

As already stated Live Connect allows the creation of new instances of Java classes, to set and query fields of classes and their instances and to invoke public methods of these classes or instances. It should be clear, however, that Live Connect has a number of restrictions. It does not allow the definition of new Java classes or subclasses from within Java Script or the creation of Java arrays. It doesn't allow Java Script programmes to use the java.io.File class for security reasons. If Java Script programmes were allowed the use of java.io.File class they would be capable of reading, writing and deleting files on the host system, giving it the capabilities required for Internet 'viruses'. It is because of this and other security concerns that Java Script only allowed the same permissions as untrusted applets.

JAVA SCRIPT TO JAVA

Since for 'Courseware on-the-Fly' it is required that Java Script will work with Java Applets it is necessary to discuss this aspect of Live Connect here. Because Live Connect exposes the Java object for each applet on a web page it is possible to read and write public fields as well as invoke public methods of the applet. For the purpose of 'Courseware on-the-fly' it is the latter aspect of this that is of concern. Below is an example of how the Java applet could be called;

```
<INPUT type= “button” value= “unit_id” onlick =
“document.Jdbcmy.myexecute(‘section_id’, ‘unit_id’)” >
```

It should be noted here that in the <APPLET> tag that the applet is given a name attribute, ‘Jdbcmy’. The value of this name attribute is then used as a name of a property, (the applet), in the document object. It now becomes possible to refer to the applet as ‘document.Jdbcmy’. The Java Script then can call methods belonging to the applet as ‘myexecute’ is called above.

JAVA TO JAVA SCRIPT

We must now consider how Live Connect enables Java to use Java Script methods. This is accomplished mainly by the use of the netscape.javascript.JSObject class, which would allows the applet in ‘Courseware on-the-fly’ to write out HTML on the fly !

An instance of the netscape.javascript.JSObject class acts as a wrapper around a single Java Script object. The class defines methods that allows the reading and writing of property values and array elements of the Java Script object as well as invoke methods of the object. A Java applet must first obtain a JSObject representing the browser window, or frame, in which the applet appears in order to make interaction with the Java Script objects possible. There isn’t a constructor method defined by the JSObject class, therefore, the static getWindow() method must be called. When this method is passed a reference to an applet itself, it returns a JSObject that represents the browser window containing that applet. Once the JSObject that refers to the ‘root’ window is obtained, instance methods of the JSObject can be used to read and write values of the properties of the Java Script object that it represents.

Besides reading and writing properties from Java Script objects, the JSObject class also allows you to invoke methods of Java Script objects. This is the characteristic of the JSObject class which is of most concern for ‘Courseware on-the -fly’. It is this characteristic that allows the writing of HTML on-the-fly from the Java Applet. The JSObject call() method invokes named methods of specific Java Script objects and passes specified arrays of Java objects as arguments to the methods. Care must be taken here as it is not possible to pass primitive Java values as arguments to a Java Script method, the primitive Java object’s corresponding Java object types must be passed instead.

```
JSObject win = JSObject.getWindow(this);
String[] args = {"Loading"};
win.call("alert",args);
```

(Example of how a Java Applet might open an alert window !)

Another more useful JSObject method is eval(), which executes a string that contains JavaScript code. The method is much more useful than the call() for HTML on-the-fly, because a string representation of the data types being passed can be used removing the necessity to convert Java primitive types to their corresponding object types. This can be proved by the code below which is used to print out a statement to the HTML window.

```
JSObject win = JSObject.getWindow(this);
```

```
win.eval( “parent.frame[1].document.writeln( ‘my message’)”);
```

Without the eval() method it would require five extra lines of code per ‘write’ statement.

```
JSObject win = JSObject.getWindow(this);
JSObject parent = (JSObject) win.getMemeber(“parent”);
JSObjec frames = (JSObject) parent.getMemeber(“frames”);
JSObject frame1 = (JSObject) frames.getSlot(1);
JSObject document = (JSObject) frame1.getMember(“document”);
Object[] args = { “my message” };
document.call(“write”, args);
```

One further requirement must be satisfied before a Java applet can be allowed to interact with Java Script. This is a security consideration where applets are not allowed to use Java Script unless the author of the web page, who may or may not be different that the developer of the Applet, gives explicit permission to the applet. To give this permission the MAYSCRIPT attribute must be used in the <APPLET> tag in the HTML file.

```
<APPLET codebase="." archive="/jdbc/lib/classes111.zip"
code="Jdbcnew.class" name="jdbcmy" width=2 height=2
MAYSCRIPT=yes></APPLET>
```

4.6 CONCLUDING REMARKS

With Netscape’s Live Connect allowing inter-communication between Applets and Java Script we have Applets capable of writing HTML on-the-fly. With Oracle’s JDBC thin driver allowing communication between the Database and the Java Applet content can be queried from the Database. When both of these are combined ‘Courseware on-the-fly’ becomes possible.

In the next chapter the overall design and implementation of a course for ‘Courseware on-the-fly’ will be examined. It will involve considering how ‘Courseware on-the-fly’ satisfies the requirements for developing a computer aided course.

5. EXAMPLE COURSEWARE

This chapter will involve an in-depth look at the design and implementation of Web enabled courseware. The example taken will be C++ pointers, the course section which was implemented for evaluating ‘Courseware on-the fly’.

Many third level Computer Science and Engineering students have difficulty with C++ pointers. Pointers are one of the most powerful features of any language especially C++. They are also among C++’s most difficult and challenging capabilities to master. Pointers enable programs to create and manipulate dynamic data structures such as linked lists or queues. They also allow programs to simulate call-by-reference. The section on C++ pointers offered by ‘Courseware on-the fly’ includes basic pointer concepts such as declarations and more advanced uses like creating and using dynamic data structures.

5.1 GUIDELINES AND REQUIREMENTS

Paperbound documents, books and lecture handouts have been the traditional method of supporting third level teaching. The purpose of this thesis is to present an approach of ‘Courseware on-the-fly’ which is close to the traditional style of lecturing and which is of a quality and standard far superior than books or lecture videos. This is achieved by the course content provider, the lecturer, concentrating on how to present certain topics without having to depend too much on mastering the systems technology.

There are a number of guidelines which should be followed when developing courses which are to prove to be successful WWW based courses. The presentation of the course material should support;

1. The four basic principles of teaching.
2. Student Testing.
3. Student Motivation.
4. Student’s Cognitive Structure.
5. Student Collaboration.
6. Courseware Ease of Use.

5.1.1 The four basic principles of teaching.

The presentation of the course material should support the four basic principles of teaching which are [WAD97];

(A) Media Richness. This is where the course environment uses various forms of media to facilitate the transfer of knowledge to the student.

(B) Interactivity. The term interactive learning can be difficult to define clearly. All learning can be said to be active since the student is required to 'do something' [TUC92]. A minimum condition for interactive learning to occur is that the student should be engaged in a theory building activity where he/she is required to actively reflect on what is being taught. Interactivity can be defined therefore as a learner's ability to respond in some way to material and to obtain feedback from these responses. This notion of interactive learning recognises the link between student and the material to be studied. Interactive learning then contains the notion of intra-active learning where the student actively integrates new and existing knowledge. Interactivity when used correctly increases learning effectiveness considerably. When examining how course material is to be presented it should be clear that the medium used is interactive and user friendly. Within educational technology three types of interactivity which can be supported are identifiable.

- The first form of interactivity is Functional interactivity which is the most important form of interactivity. This is how the students and the system interact to accomplish goals.
- The second is Adaptive interactivity which involves students adapting the course content to suit their needs and preferences.
- The third, most basic form, is Navigational interactivity. This is where students interact with the system so as to traverse the course in the way they wish.

It should be clear that for interactive learning to occur the course must be so designed that focus is placed on how some of the responsibility for successful learning falls on the learner. This does not mean that the blame for poor teaching is shifted on to the students, but that, in accordance with the constructivist theory, the students actually do the job of learning rather than having knowledge simply 'dumped' on them. There are a number of factors which determine system interactivity [GREY87];

- How the interactivity is achieved ?
- Where does the course material contain these interactions ?
- Is the interaction too complex or too simple ?
- Are all student reactions anticipated ?
- Is the student feedback detailed enough without being too detailed and therefore confusing to the students ?
- Do the students have control over the learning process ?
- Is an interaction appropriate or wasteful ?

(C) Timely Responsiveness. This is a requirement where students are provided with the course material when it is needed to further learning. Care must be taken here to ensure that the student is neither starved of the information required or suffers from 'information drowning'.

(D) Course Organisation. This is the primary responsibility of the course director ! The course material must be organised into separate sections such as a beginner's section on

C++ pointers for example. These sections then must be organised into separate units such as;

- Pointers: What and Why ?
- Pointer Arithmetic.
- Linked Lists.

5.1.2 Student Testing

Students should be asked to complete mini tests which then should be corrected on-the-fly. This supports student's retention of information and knowledge, and conforms with the behaviourist view of education where repetition is one of the major keys of retention. Although behaviourism is by no means the best educational theory, some of it's ideas are useful when used properly.

5.1.3 Student Motivation

The medium used should help to motivate the student into using it more frequently than he/she would normally do. The students attitude, which can be defined as the readiness to respond in a particular way when confronted with a particular objective, affects motivation but is also in turn affected by prior learning experiences. Motivational drive can affect the learner's cognitive approach to learning. The motivation used can be of two forms;

(A) Intrinsic motivation, which is the most important form, involves the student having interest in what is being learned. This type of motivation correlates with a preference for constructivistic learning. Bruner [BRU60] claimed that the best way to generate intrinsic motivation is to render it worth knowing, "to make the knowledge gained useable in one's thinking beyond the situation in which learning has occurred [to effect transfer]". In simple terms this means that something will be more quickly learned if it is placed in the correct context.

(B) Extrinsic motivation which is the need for qualifications or the fear of failure, would in general be associated with rote-learning which is of no great advantage to anyone concerned.

5.1.4 Student's Cognitive Structure

The cognitive structures of the student should be supported within the learning environment. The computer can provide the functionality necessary to help develop these cognitive structures and skills [TUC92]. The facilities which help this development include; bookmarks, responses to questions and queries from/to the student as well as periodic review of the course material.

5.1.5 Student Collaboration

Student collaboration should be supported to help promote learning. Collaboration is the process whereby students work closely with other students or the lecturer for their mutual benefit. There are a number of techniques which can be used to promote collaborative learning [DILL95].

- Shared Cognitive Load; This is where the problem solving process involves a number of different individuals sharing the burden.
- Social Grounding; This involves the teacher monitoring the listeners understanding so as to ensure that the students don't become completely lost.
- Conflict; This happens where a problem solving exercise causes disagreement within a group as to the best method of solving a problem.
- Alternative Proposal; This is where a number of different solutions to a problem are proposed by different people and then discussed within a group.
- Explaining; If one student is required to give an explanation of a problem to another student both students tend to gain in knowledge from the experience.
- Internalisation; Group conversation promotes understanding on the same level among group members. This type of learning is known as Internalisation.
- Appropriation; This involves two people working together by providing input into each others work.
- Mutual Regulation; This involves students justifying their answers to other students or to the teacher.

Student collaboration can be supported by technology such as e-mail, where the student can communicate with other students or the course director.

5.1.6 Courseware Ease of Use

A good student user interface will facilitate the student's interaction with the course material without the hindrance of having to deal with complex technical considerations of the interface. This issue also comes under the umbrella of the previously discussed 'Interactive Learning'. The user interface must be clear and easy to use. If the user interface is not designed properly with the education theories and requirements in mind the course material offered will be of no use to the students. If students cannot figure out how to access the course material easily then they lose interest and the course loses all effectiveness.

5.2 THE COURSE MATERIAL

The course material designed for 'Courseware on-the-fly' must focus on two important issues, course material re-use and re-purpose. Re-use is to use again whereas re-purpose is to use again on an application for which it wasn't originally designed. The distinction between reuse and re-purpose can often be a bit blurred. Reuse can primarily be seen in the course material designed here, C++ Pointers, where units can be viewed by different students. Re-purpose can be understood then as 'reusing' the same course content in a different way. An example of this is where the Linked Lists unit is reused and re-purposed for making up different course sections (an Intermediate section and an Advanced section are examples of this). This can be seen with practically all the units which are re-used to make up different sections of the course. The course could, however, be re-purposed if the course material , or parts thereof, was to be used as a course module in a new course for engineers (taking the assumption that it was originally designed for computer science students).

As stated before the course material chosen as an example for 'Courseware on-the-fly' was C++ pointers. Since 'Course Organisation' is one of the four key principles of teaching great care was taken when organising the course material into units and then deciding which units make up which section. Student modelling principles were used to facilitate good course organisation.

5.2.1 Student Modelling

There are three main reasons for gaining a user model upon which to base the course organisation [STA96];

1. There is less of a risk of information overload. Information overload represents either too much information being presented to the student or the incorrect information being presented.
2. The benefits of a course for a student are much greater if the course is somewhat tailored to his/her requirements.
3. If there is a personalisational feel about the course the student may feel more confident and relaxed about using it.

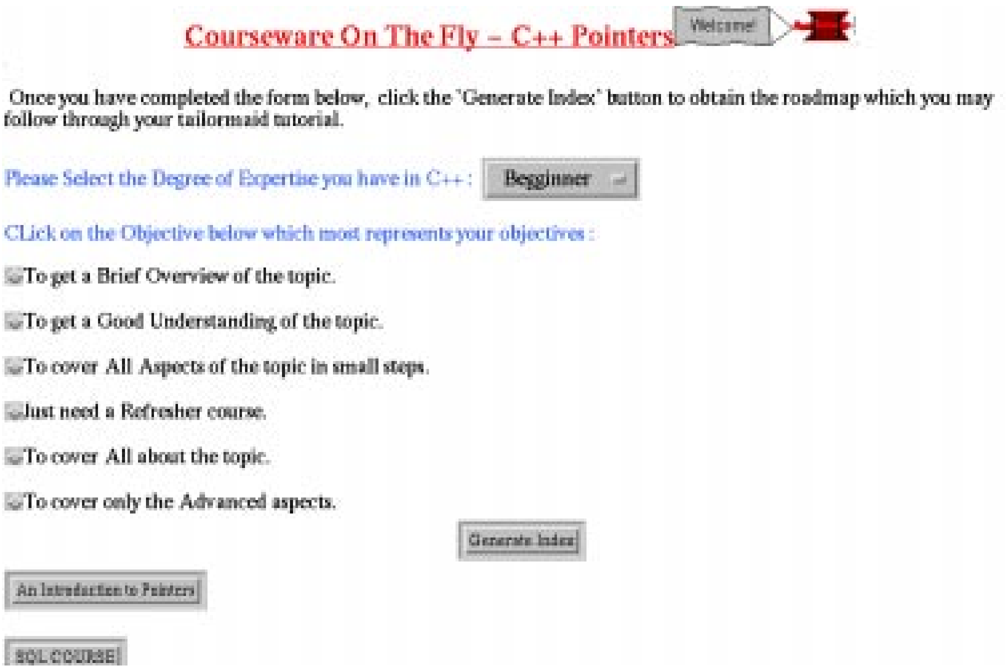
In order for an accurate student model to be constructed it is important to elicit the following information about the student(s) taking the course [WHI98].

- What previous knowledge the student has on the area of study.
- What does the student hope to have achieved upon completion of the course.

Is it also beneficial to decide whether or not to provide the students with control over of the units which they cover, or is it best to direct the student as to which units they cover, (Directed Learning). With 'Courseware on-the-fly' this is achieved by allowing some students, depending on the student model, to select extra units which they can cover within the period of their course. Traditionally knowledge acquisition was the job of a knowledge engineer who acted as an intermediary between the computer system and the students [WHI98]. 'Courseware on-the-fly' uses automated knowledge acquisition to reduce the role of the knowledge engineer and to get a more recent caption of the students' understanding.

The student model is constructed when the students first commence the course. A number of questions are posed so as to get an adequate feel for the students' previous knowledge and goals. Once the student model is created the students are then presented with the course 'Roadmap' which they are expected to follow. The students may however be refused any personalisation if they are told by the lecturer to select one of a number of ‘canned’ course roadmaps also available. The canned roadmaps present the lecturer with the ability to have an entire group or class follow exactly the same syllabus for examination purposes.

EXAMPLE STUDENT MODEL



Courseware On The Fly – C++ Pointers

Once you have completed the form below, click the 'Generate Index' button to obtain the roadmap which you may follow through your tailormaid tutorial.

Please Select the Degree of Expertise you have in C++ : Begginner

CLick on the Objective below which most represents your objectives :

○ To get a Brief Overview of the topic.

○ To get a Good Understanding of the topic.

○ To cover All Aspects of the topic in small steps.

○ Just need a Refresher course.

○ To cover All about the topic.

○ To cover only the Advanced aspects.

(Diagram 4.1, Questions asked to create student model)

The student model created for use with ‘Courseware on-the-fly’ is based on the students classifying themselves as being at one of three levels of competence.

- Beginner
- Intermediate
- Advanced

Once it is determined within which stream the students are classified, the next problem that needs to be addressed by the students is their expectations upon completion of the course. Take for example a students claiming to have intermediate expertise with C++ and just requiring a refresher course on C++ pointers. The students would be presented with the following units as being compulsory for completion;

- ‘More about Pointers’
- ‘Pointer Arithmetic’
- ‘Pointer-Arrays as one’
- ‘Linked Lists’

In addition to these compulsory sections the students are offered two optional units which they can select if they so desire;

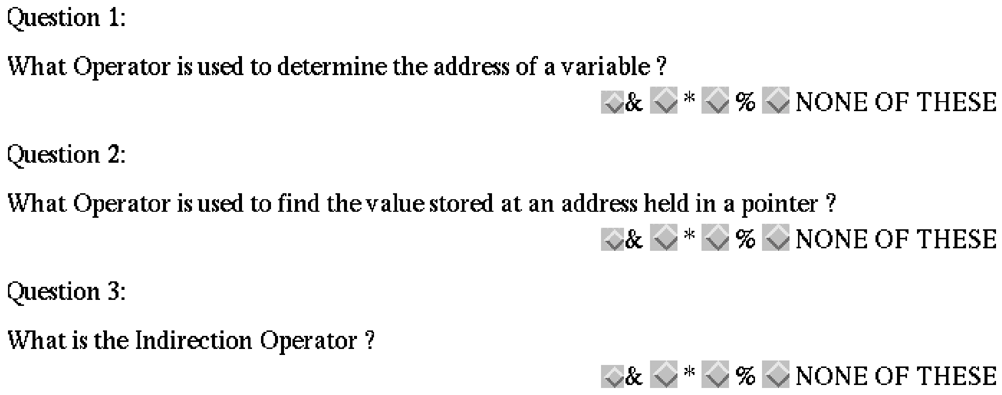
- 'Pointers: Creating and Using'
- 'Dynamic Memory'

If any of these optional units are selected by the students it may be necessary to reorganise the road-map index. This reorganisation occurs to ensure that the units in the students course index are in the correct order.

5.3 STUDENT EVALUATION / TESTING

'Formative Evaluation' is carried out throughout the course to provide student evaluation / testing. Questions are integrated within the different units which the students are required to complete. There are a number of reasons why this form of evaluation is beneficial;

- The students learning progress can be monitored.
- Corrective feedback can be supplied here to correct any misconceptions on the part of the students.
- Students can be refocused so that the most important issues are emphasised.



Question 1:

What Operator is used to determine the address of a variable ?

○& ○* ○% ○NONE OF THESE

Question 2:

What Operator is used to find the value stored at an address held in a pointer ?

○& ○* ○% ○NONE OF THESE

Question 3:

What is the Indirection Operator ?

○& ○* ○% ○NONE OF THESE

(Diagram 4.2, Example of some of the questions asked)

These questions are corrected on-the-fly. The students are allowed to re-try the questions as many times as they wish or they can give up and view the answers. This process has been referred by behaviourists as 'Drill and Practice'. Drill and practice is useful for students to refresh their existing or newly learned skill/knowledge. There is, however, a slight difference between the questioning here and the traditional 'Drill and Practice'. With 'Courseware on-the-fly' the students decide how often they complete the questions posed. This removes the boredom which could be caused by students continually repeating questions which they consider basic.

The questions which can be posed to the students can take many different forms. The quality of the questions depends not only on what forms they take but how they are tuned and used appropriately. The two main types of question which were chosen for the example course, C++ pointers, were True-False questions and multiple choice questions. True-False questions can be advantageous since it generally takes less time and effort to construct and correct them. They do however have the drawback that there is 50:50 chance that the student may have guessed the right answer. Multiple choice questions on the other hand have many more advantages;

- Student writing is minimised allowing for a large amount of course material to be covered in a short amount of time.
- Multiple choice questions can represent learning objectives for simple memorisation tasks as well as cognitive manipulations.
- If about five alternatives are given the student's ability to guess the correct answer is significantly reduced.
- Incorrect alternatives can be used to detect common errors by students.

With both types of questions there can be a number of common guidelines identified so that students obtain most value from completing the questions. These guidelines include;

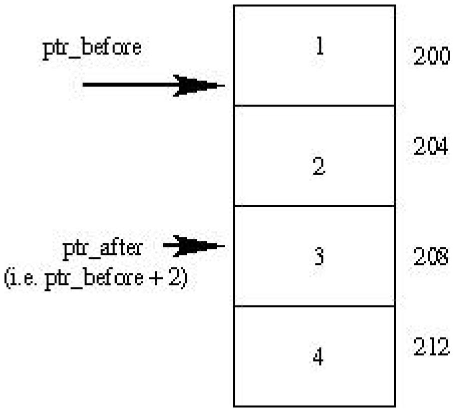
- Ensuring that the instructions and questions posed are clear and concise.
- Visual aids such as charts and tables wherever possible.
- Providing only one correct or at least clearly better answer.
- Ensuring that all alternatives are plausible. This can be done by including common misconceptions as alternatives.

5.4 COURSE APPEARANCE

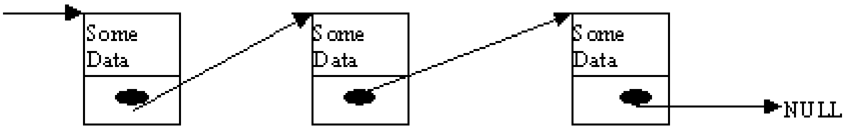
The issue of course appearance is often one of the most important issues for a course director. The reason for this is that if the course does not look interesting or appealing then the student won't be motivated to do well in the course. This is where the issue of media richness is of concern.

5.4.1 Media Richness

There are a number of different media which could be used to facilitate student learning. The main types of media used for the course on C++ pointers included Text, Images, and of course the Internet. Text is the most common form of presenting material to a student and in this case refers to the alphanumeric characters presented on the computer screen. This is the form used to provide the tuition and the details of the course. Images and Visual aids can be used to focus the students attention on something or to provide greater clarification. The Chinese proverb ' A picture paints a thousand words ' rings true here. Examples of it's use within the course designed here can be seen right throughout the course.



(Diagram 4.3, A Memory Map)



(Diagram 4.4, A Linked List)

The Internet is the medium upon which 'Courseware on-the-fly' resides. It provides the opportunity to the students to access the course whenever they desire. This medium was discussed earlier in chapter 3. The use of e-mail and how it satisfies the need for student collaboration was discussed earlier in this chapter. Other media which could have been used here include Audio and Video. Audio and Video recordings in conjunction with the Internet can be played for the student so as to provide demonstrations and clearer presentation of material.

5.4.2 Student Motivation and Interactivity

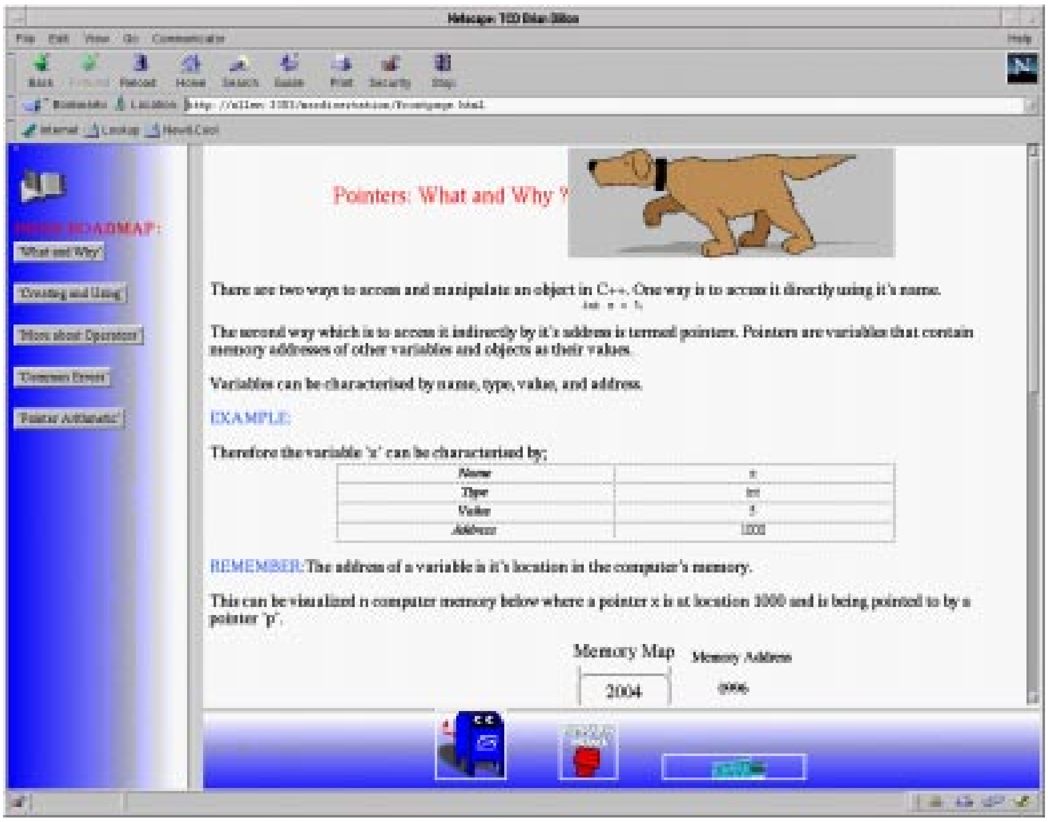
The issues of student motivation and interactivity although distinct can be discussed together. Student interactivity provides the students with some interest in the course and therefore can serve to motivate the students to do well. Student interactivity was supported in the course developed here by posing questions to students to determine which course they should take as well as allowing them to select additional course units which they would like to cover. Interactivity is also supported through the facility of asking review questions throughout the course. Students are also provided with the facility to view the execution of some of the C++ code, through the use of Java Applets which simulate the code execution.

All the above interactivity aids serve to promote interest in the subject to be learned and therefore serves to promote intrinsic motivation. Students interest must also be generated and held by the use of language which is neither too patronising nor too complicated. This involves designing tuition on problems which is easily understood by the students while still holding their interest in the problems. The graphical aid of images and diagrams can be used to overcome this problem.

5.4.3 Overall Course Appearance

One of the most important things for a computer based course to be successful is the courses aesthetic appeal. Students tend to be more interested in a course which is visually effective and easy to use. This section will discuss the overall layout and appearance of the user interface for 'Courseware on-the-fly'.

It was decided that frames would be used to display the course material since they provided a way of displaying multiple pages within a single window. Frames allow a single browser window to be divided horizontally and / or vertically into individual sections that each displays a separate HTML document. Frames then allow Courseware on-the-fly to not only display the course material but also the roadmap and other related buttons simultaneously.



Pointers: What and Why ?

There are two ways to access and manipulate an object in C++. One way is to access it directly using it's name.

int x = 5;

The second way which is to access it indirectly by it's address is termed pointers. Pointers are variables that contain memory addresses of other variables and objects as their values.

Variables can be characterised by name, type, value, and address.

EXAMPLE:

Therefore the variable 'x' can be characterised by;

Name	x
Type	int
Value	5
Address	1000

REMEMBER:The address of a variable is it's location in the computer's memory.

This can be visualized n computer memory below where a pointer x is at location 1000 and is being pointed to by a pointer 'p'.

(Diagram 4.5, Course Frames)

The first page of the interface provides the students with the ability to select the course they wish to take. The layout here is illustrated above. The main frame is where the student details are taken and the course material is displayed. Buttons are provided along the bottom so that the students can ;

- E-mail the course director / lecturer about the course.
- Return to the main page whenever they wish to change the course they are taking.
- View the course instructions whenever they require to do so.

The vertical frame on the left-hand side of the screen is where the students' roadmaps are displayed. The students can navigate through the course by clicking on each 'unit button' in turn. As stated earlier the students may be provided with optional units which can be selected and these selected units are added to the bottom of the roadmap in the index frame. When the students are happy that they have selected all the additional units that are required then the roadmap can be reordered once the 'reorder' button is depressed.

5.5 CONCLUDING REMARKS

This chapter has dealt with all the necessary design issues which must be satisfied when designing a computer aided course. An example course, C++ Pointers, was explained and discussed so as to illustrate how 'Courseware on-the-fly' can be used to solve the design issues mentioned. The next chapter will show how the back-end architecture was designed and implemented.

6. SYSTEM ARCHITECTURE

In this chapter the design of the back end supporting architecture will be detailed and explained. This chapter follows on from chapter three where the different technology choices were discussed and chapter four where the design and implementation of a course was explained.

6.1 DATABASE DESIGN

The first issue that needs to be examined here is the storage of the course material. In chapter three it was explained that the choice had been made to choose a database in preference to the traditional flat file method of storage. Oracle databases are probably the most popular if not the best relational databases in production. The chosen database for 'Courseware on-the-fly' was Oracle 8 with NET 8 for use with networks. This is a Relational database which means that it is organised according to the relational model of data which was introduced by E.F. Codd in 1970. It has the advantage that it provides a conceptually simple view of the system's data structure while having a solid mathematical base. The relational data model defines three aspects of the system [ELM94]:

- Architecture: This describes the way data objects are structured.
- Data Operations: This describes the methods which can be used to manipulate the data objects.
- System constraints: This specifies the rules that govern the operations and ensure that the result of each operation is as expected.

6.1.1 Database Architecture

The relational model represents the database as a collection of relations, called tables. The table is the most common database object defined, it consists of a two dimensional matrix. Each row in the table represents a collection of related data values. Examples of tables required for 'Courseware on-the-fly' are;

- Course makeup: This describes what Sections, e.g. 'Intermediate Refresher' section, are put together to make a course such as C++ pointers.
- Section makeup: This represents how each Section is made using course units, such as 'Linked Lists'.
- Unit makeup: The next information required is on the construction of units from individual pagelets. Pagelets are parts of a page, such as paragraphs and images, which when combined in certain way(s) make course unit(s).
- Pagelet metadata: This table is used to describe certain meta-information required by the course director / co-ordinator to keep track of the various pagelets.

- Pagelet data: This table contains the actual data, course material, which make up the course in the manner specified by the above tables.

Each column in a table represents an attribute, a single data value, of the entity which the table describes. An example of an entity attribute is 'Section_id' which is used to help distinguish between different course sections. By definition all elements of a relation are distinct, therefore every row within a table must be unique. There must be a 'primary key' of attributes designated, where no two rows in the table has the same combination of values for those attributes. Take the table which contains the pagelet data, the primary key contains only one attribute 'Pagelet_id' to uniquely identify every row in the table. It should be noted here that the perfect attribute should be unique in all tables in the database unless it is used as a key and it should not be possible for it to be broken into smaller components [ASH98].

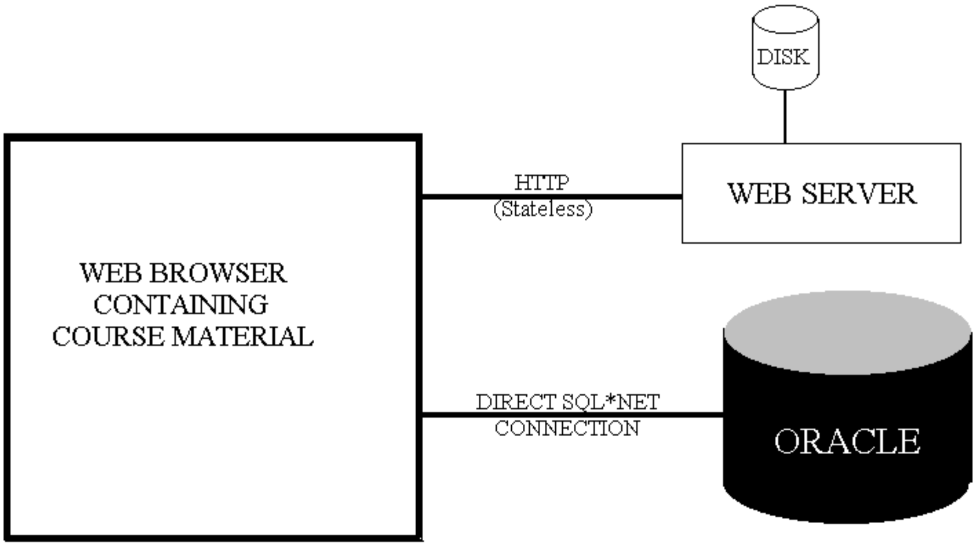
6.1.2 Data and Field Types

The next issue to be considered was what kind of data would go in each field of the table. There are five main types of fields which can be used with databases [ELM94]:

- Number; This type can be used in mathematical operations or for providing information about the ordering of the table rows.
- Time/Date; Any variation of time and date can be stored in this type of field.
- Character; The character data-types are used to store and manipulate words, strings and free-form text. They are alphanumeric data strings stored as byte values in the database on the assigned character set, e.g. seven-bit ASCII. In Oracle there are two main character data-types, CHAR and VARCHAR. The CHAR data-type is used when storing relatively short fixed length character strings whereas the VARCHAR data-type is used when storing variable length character strings which can hold alphanumeric characters up to a maximum length.
- LONG; The LONG data-type in Oracle is used to store very long, variable length character strings, such as pagelets.
- Binary; The binary data-types are used to store raw byte-oriented binary data. This could include a graphic image or a sound sample.

It should be noted here that not all the information needed for the course is stored in the database. Although there are provisions, as shown above, for images to be stored in the database, it was found that to facilitate the student copying the image it was best not to store the explanatory images in the database. Instead the images are stored on disk. Another complication which arose was that Java Script programmes and Java Applets cannot be stored effectively in the database so these too needed to be stored on disk. The data which is stored on disk is referenced in the database so that it can be retrieved on-the-fly with the rest of the course.

One major advantage of having additional data stored on disk, and only referenced by the database, is that example programmes can be changed easily by the course director. This facilitates easier update of the material since there is only one location where the data must be updated. Take for example a computer programme which the lecturer wishes to update. If the material was contained in the database it would need to be updated on disk in any case, since the compiler would not be accessible from the database. The changes would then require mirroring in the database requiring some additional database programming. If the database only referenced the material, the programme would only require changing in one place, on disk. Note that this does not apply to the regular course material as it would not require any recompilation of material ! The system architecture so far can be viewed as;



(Diagram 5.1, Storage of Course Material)

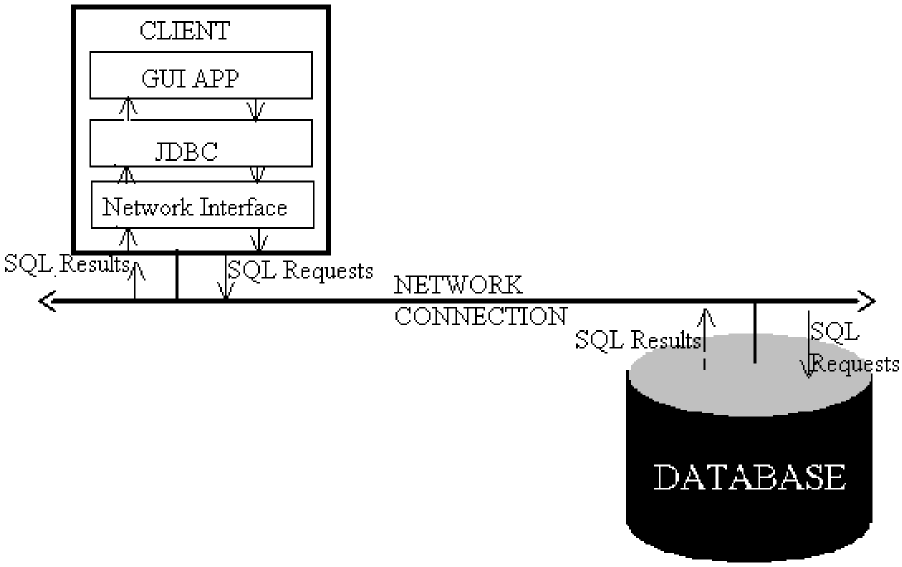
6.2 DATABASE ACCESS

Once the database has been designed it is important to look at how this database is to be connected to the Web. In chapter three we spent some time discussing Java Database Connectivity, JDBC, and choosing the best available driver for connection. The type of driver which was chosen was the JDBC-thin driver developed by Oracle. In this section we will discuss this driver and how it helps for the system architecture for ‘Courseware on-the-fly’.

JDBC, as explained in chapter three, is a set of specifications that define how a programme written in Java can communicate and interact with databases [SIP98]. The JDBC API defines how this communication should take place and how the database and application interact together. This includes such things as;

- How an application opens a connection to the database.
- How the application then communicates with the database.
- How SQL statements are executed and results retrieved.

6.2.1 Data Flow Model

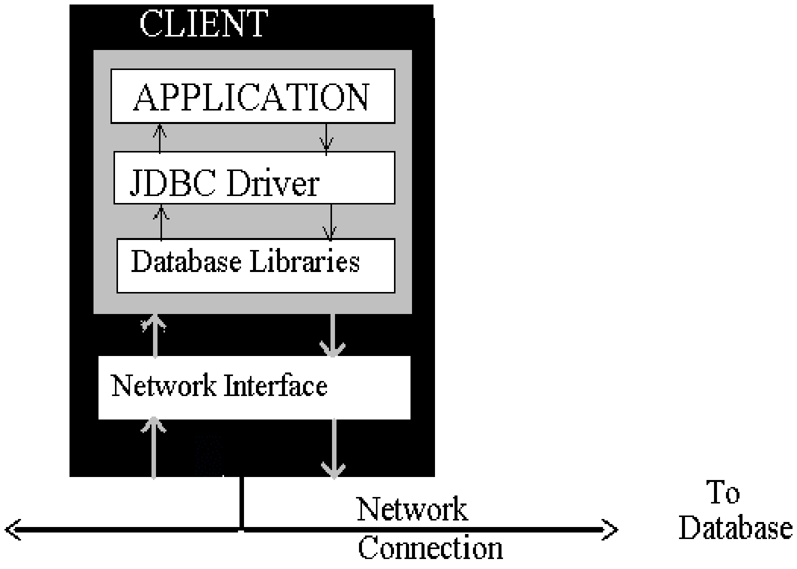


(Diagram 5.2, JDBC Data Flow)

The JDBC data flow model can be seen in the diagram above. All interaction with the database in implemented through 'raw' SQL statements. The Graphical User Interface, GUI, components are therefore imported from the Java Abstract Window Toolkit, AWT.

6.2.2 JDBC Driver

Being an interface the JDBC API defines two things. Firstly it tells the Application programme what is expected from the database, i.e. it defines how information can be presented to an application. Secondly it needs to define what the database expects from the application. The JDBC API therefore defines the common ground between database and application. The JDBC driver is at the heart of all this.

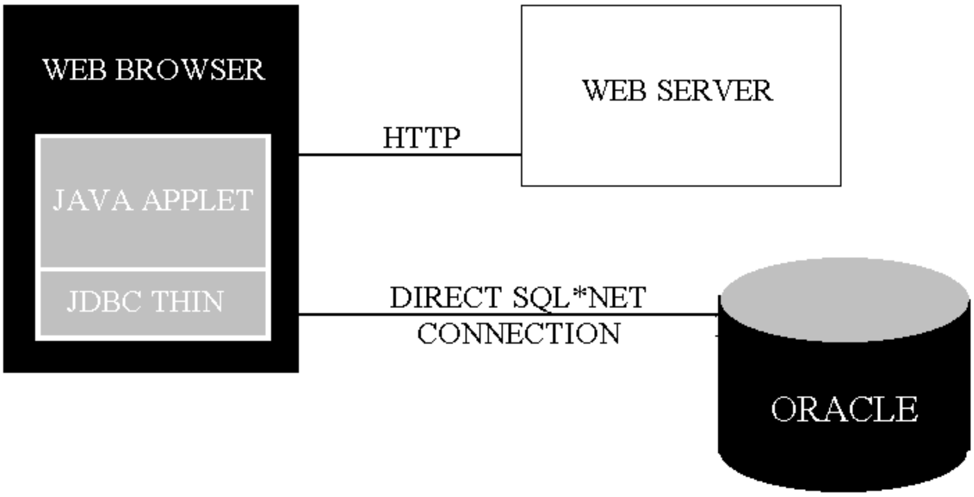


(Diagram 5.3, JDBC)

The JDBC driver is responsible for ensuring that an application has consistent and uniform access to any database. It is also held responsible for ensuring that any requests made to the database, from the application, are meaningful. In the diagram above it can be seen that the JDBC driver is central here, accepting instructions from the application (this corresponds to one set of arrows) and making the translations required to forward the access requests to the database. The responses are dealt with the same way. The database sends back the data so the JDBC driver can translate the data into Java data format and then forward it onto the application.

The JDBC-thin driver can be downloaded with Java applets resulting in performance improvements over other JDBC drivers. In ‘Courseware on-the-fly’ the applets involved in making the course selections and course material retrieval contain the JDBC-thin driver. The Java applet and the JDBC-thin driver are downloaded into the browser from the web server. The JDBC-thin driver then establishes a direct connection with the Oracle database server over Java sockets. Since the JDBC-thin driver provides it’s own lightweight implementation of a TCP/IP version of Oracle’s SQL*/Net8 protocol, it’s communication from/to the database is stateful.

The JDBC-thin driver can be used in a number of different configurations. With the current example of ‘Courseware on-the-fly’ the JDBC-thin driver does not need to deal with firewalls and for convenience the database and web server are on the same machine. If this was not the case then the applet which contains the JDBC-thin driver would need to be signed for security reasons. The architecture which is used is illustrated below.



(Diagram 5.4, JDBC-thin Driver Architecture)

With the above architecture and the previously explained database architecture ‘Courseware on-the-fly’ gains the facility of storing and retrieving the course material and meta-information on how it is to be organised. The problem now is how can this material be displayed in a graphical and meaningful way on the Web Browser. Without further development the information could only be displayed within the Java Applet which does not facilitate structuring of information on-the-fly. This is not, however, the only problem ! A question which also needs to be addressed is, how does the student use the applet containing the JDBC classes ? It is not desirable to expose students to the Java applet and it’s intricacies. One of the purposes of ‘Courseware on-the-fly’ is to make using course material authored on-the-fly as painless as possible for both the students and the course developer. The solution to these problems was introduced in chapter three, Netscape’s Live Connect which facilitates communication between Java Script and Java Applets.

6.3 HTML ON-THE-FLY

HTML on-the-fly is what this thesis is mostly concerned with, it can be facilitated through the use of Java Script and Live Connect.

6.3.1 Java Script

In Chapter three Java Script was chosen as a good method for writing out HTML on-the-fly. Java Script as developed by Netscape is a relatively general purpose programming language, it has a number of different capabilities which include it’s ability to: [FLA97]

- Control Document Appearance and Content; Through the use of the Java Script document objects ‘write()’ method arbitrary HTML can be written into a document as it is being parsed.

- Control the Browser; There are several Java Script objects which allow control over the behaviour of the browser. An example is the Window object which supports methods used to pop up dialogue boxes which display simple messages.
- Facilitate Document Content Interaction; The Java Script document object, and the objects contained within, allow programmes to read and sometimes interact with portions of documents.
- Allow User Interaction; An important feature of Java Script is it's ability to define 'event handlers' which are executed when a particular event occurs.
- Read and Write Client State with Cookies; 'Cookies' are Netscape's term for small amounts of state data stored permanently or temporarily.
- Interact with Applets; As stated earlier Java Script can interact with Java applets that are running in the browser.

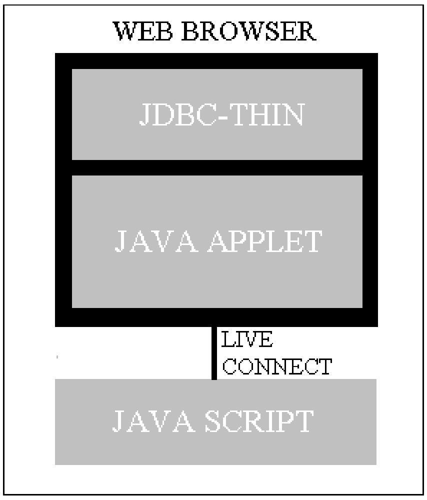
Java Script is used in 'Courseware on-the-fly' for correcting questions on-the-fly and for accessing applets to enable authoring of course material on-the-fly.

6.3.2 Live Connect

Live Connect makes it possible to design Web pages which have a new and more dynamic level of interactivity. This 'dynamic level of interactivity' is achieved by allowing;

- Netscape Navigator plug-ins to interact with Java Scripts running on the same HTML page.
- Java applets to communicate with Java Scripts and Java Scripts to call Java Applets methods.

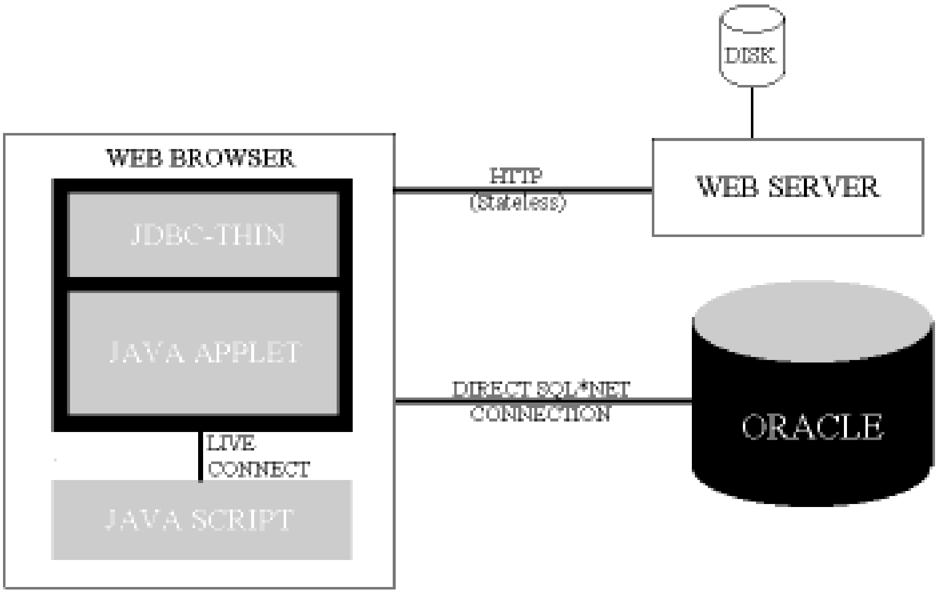
With 'Courseware on-the-fly' it is the later issue which is of concern. To insulate the student from the Applet, Java Script is used to gain the information required for course organisation or retrieval of course material from the database. Since JDBC works with Java Applets and Java Script, the Java Script programme must call the required method in the Applet to do the necessary communication with the database.



(Diagram 5.5, HTML on-the-fly)

Now that it is possible to contact the Java Applet and retrieve valuable information from the Oracle database using JDBC, it is necessary to consider how does the Java applet write out HTML on-the-fly ? If Netscape version 4.0 is used then Live Connect solves the problem by allowing the Java Applet call Java Script functions within itself. The Java Applet must firstly get the Web browser window handle before it is allowed access to the Java Script functions. As stated earlier Java Script can write to a document using it's 'write()' method. It therefore, follows on that a Java Applet can write out HTML on-the-fly by obtaining the browser window handle (using the JSObject class's getWindow() method) and then calling Java Script's 'write() method !

6.4 PUTTING IT ALL TOGETHER



(Diagram 5.6, Full System Architecture)

The finished system architecture for 'Courseware on-the-fly' can be seen above. Netscape's Live Connect is used by Java Script programmes to call the relevant Java Applets functions. These Applets in turn can use Oracle's JDBC-thin driver to access the Oracle database and retrieve any information required. Once this information is retrieved it is displayed using Live Connect to call Java Script methods to write out HTML on-the-fly. This HTML on-the-fly could also contain images, located locally on disk, or Java Applets required for further information processing / retrieval.

6.5 CONCLUDING REMARKS

This chapter has dealt with the underlying architecture for ‘Courseware on-the-fly’. With this architecture and the course, C++ pointers, from chapter four it can be clearly seen how ‘Courseware on-the-fly’ should be implemented. The next chapter will discuss the implementation and integration details which arose.

7. IMPLEMENTATION

In this chapter the implementation details of ‘Courseware on-the-fly’ will be discussed. This implementation is only an example of what can be achieved using the technologies and educational theories previously mentioned. The first section will discuss how the database was organised.

7.1 DATABASE ORGANISATION

This is one of the important issues in concern with ‘Courseware on-the-fly’. If the database is not correctly organised then it becomes impossible to select course road-maps on-the-fly which students can use to traverse their personalised course. The database used in this implementation contains five tables of information.

- Course makeup
- Section makeup
- Unit makeup
- Pagelet meta-information
- Pagelet information

These tables and their contents are discussed below.

7.1.1 Course makeup

Course_id	**Subject**	**Section_id**
C++	C++ Language	Ptrs
SQL	Student SQL	Sql

(Diagram 6.1, Cross-section of course-makeup table)

This table is organised as seen above. It is used to list all available courses with ‘Courseware on-the-fly’. At the moment there are two courses being offered, one on C++ and one on SQL. The ‘course_id’ column as it’s name suggests gives the name of the course(s) offered. The ‘subject’ attribute allows the course director to place a description of what the course is about. The ‘section_id’ distinguishes what sections make up each course. Take for example C++, this would have a section on C++ pointers (which was implemented) but would also have sections on functions and arrays among others. The ‘course_id’ and ‘section_id’ make up the primary key, meaning that together they uniquely identify all the rows in the table.

7.1.2 Section makeup

Section_id	**Subject_matter**	**Ordering**	**Unit_id**
Ptrs	Beginner Overview	1	What and Why
Ptrs	Beginner Overview	2	Creating and Using
Ptrs	Beginner Overview	3	More about Operators
Ptrs	Beginner Overview	4	Common Errors
Ptrs	Beginner Overview	5	Pointer Arithmetic

(Diagram 6.2, Cross-section of section_makeup table)

This table is used to show which combination of units comprise the various sections. The table is so organised that the course director can offer courses in the same area for differing levels of competence. It contains four columns to help organise and distinguish this information;

- 'section_id'; This is the same as for the 'course_makeup' table, here it lists the sections detailed in the table.
- 'subject_matter'; This attribute helps distinguish between sections which may have the same 'section_id' while covering different material. An example of this is if a section covered C++ pointers and within it a student could cover beginners, intermediate or advanced material.
- 'unit_id'; This attribute is used to show what units are used in each section.
- 'ordering'; This column facilitates arbitrary ordering of units within a section. It enables the course director to change how a particular section is organised.

7.1.3 Unit makeup

Section_id	**Unit_id**	Unit_dependency	**Pagelet_order**	**Pagelet_id**
Ptrs	What and Why	none	1	Ptr1
Ptrs	What and Why	none	2	Ptr2
Ptrs	What and Why	none	3	Ptr3
Ptrs	Linked Lists	Dynamic Memory	1	Ptr141

(Diagram 6.3, Cross-section of unit_makeup table)

The above table is used in constructing the units using pagelets. Pagelets as previously explained are parts of a unit such as paragraphs, images, titles and backgrounds which when 'glued' together meaningfully can construct a course unit. This table contains five attributes to describe how a unit is constructed and what pagelets are used in the construction;

- ‘section_id’ and ‘unit_id’ are the same as those in the ‘section_makeup’ table. They are used to describe the sections listed previously in the ‘section_makeup’ table.
- ‘unit_dependency’; This attribute is used to show whether a unit can be presented on it’s own or whether it relies on the student first having completed a previous unit. An example of this is the unit on ‘Linked Lists’ where, if the student is a beginner, it must be presented with the unit on ‘Dynamic Memory’.
- ‘pagelet_id’; This is used to uniquely identify each pagelet contained within the database.
- ‘pagelet_ordering’; This allows the course director to order pagelets differently, if required, in units which cover the same material.

7.1.4 Pagelet Meta_information

Pagelet_id	Topic	Difficulty	Dependency
Ptr21	Creating Using title	Beginner	none
Ptr22	Creating recap	Beginner	none
Ptr22	Initialisation	Beginner	none
Ptr23	Idirection	Intermediate	none
Ptr24	Indirection Example	Intermediate	none

(Diagram 6.4, Cross-section of pagelet_meta table)

This table is used only as an additional aid to the course director. It stores information about the pagelets. The attributes are used to describes the pagelets level of difficulty, ‘difficulty’, as well as the topic they focus on, ‘topic’. There is also an attribute for listing pagelet dependencies, ‘dependency’.

7.1.5 Pagelet Information

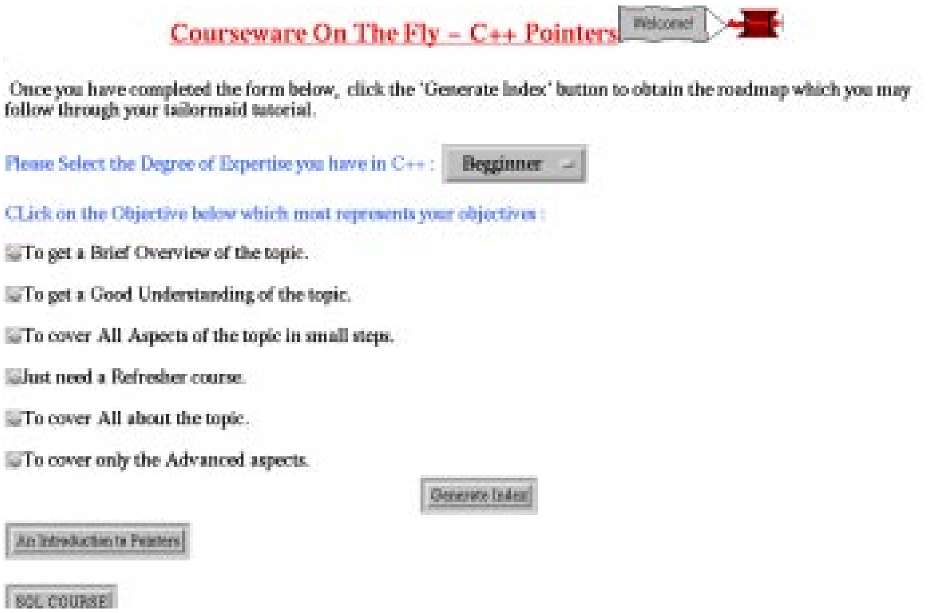
Pagelet_id	**Pagelet_info**	**Type_descriptor**
Ptr1	<BODY TEXT="#000000" BGCOLOR="#FFFFFF" LINK ="#0000EE" VLINK="#551A8B" ALINK="#FF0000"><CENTER><FONT COLOR="#FF0000"><FONT SIZE=+4>Pointers: What and Why ?</FONT></FONT><IMG SRC="dog.gif" HEIGHT=122 WIDTH=360 ALIGN=CENTER></CENTER>	Background and title

This table stores the actual pagelets themselves. It uses three attributes to describe this information. A ‘pagelet_id’ which uniquely identifies each pagelet, a ‘type_descriptor’ which describes what type of information is stored (e.g. a paragraph or title) and finally the ‘pagelet_info’ attribute which constitutes the pagelet data itself.

7.2 COURSE SELECTION

At the commencement of the course the student is asked to fill out a questionnaire, which is used to aid in the selection of the course units on offer to the students. This requires processing the questionnaire, accessing the database and then writing out the suggested road map to the student.

7.2.1 The Questionnaire



Courseware On The Fly – C++ Pointers

Once you have completed the form below, click the 'Generate Index' button to obtain the roadmap which you may follow through your tailormaid tutorial.

Please Select the Degree of Expertise you have in C++ : Begginner

CLick on the Objective below which most represents your objectives :

□ To get a Brief Overview of the topic.

□ To get a Good Understanding of the topic.

□ To cover All Aspects of the topic in small steps.

□ Just need a Refresher course.

□ To cover All about the topic.

□ To cover only the Advanced aspects.

(Diagram 6.6, The Questionnaire)

The questionnaire above uses Java Script to process the students answers. These answers divide the students into three categories; Beginner, Intermediate and Advanced. Once divided into categories the students must then indicate what their expectations are upon completion of the course. Once all this information is processed in the form it is used to choose which units of the course should be offered to the students. This is accomplished by calling a procedure ('theunits') within the applet ('Getunits'). Java Script is allowed to call this procedure within the applet because Live Connect, which was discussed in chapter 5, is used.

7.2.2 Choosing the units

The units are chosen for the student depending upon how they are classified using the student model obtained from the questionnaire. The applet ('Getunits') loads the JDBC-thin driver so that the applet has the ability to contact the database.

```
Class.forName(“oracle.jdbc.driver.OracleDriver”);
```

The connection is opened to the data upon initial loading of the HTML page containing the applet.

```
Connection conn = DriverManager.getConnection
(“jdbc:oracle:thin:@thehost:1521:database_id”, “username”, “password”);
```

Once the Applet is connected to the Database, it must create SQL statements which are sent to the database. The first SQL statement is used to select the unit_ids from the section_makeup table which match the set of criteria selected by the student in the questionnaire.

```
Statement stmt = conn.createStatement();
ResultSet rset = stmt.executeQuery ("select unit_id from section_makeup where section_id
='"+sectionid+"' AND subject_matter='"+subject_matter+"'");
```

The second SQL statement selects the optional unit_ids which students may select later if they wish to cover additional units. This allows students to personalise their road-maps through the course.

DISPLAYING THE INDEX

Once the units and the optional units are retrieved form the database by the applet, it becomes necessary for students to view the selections made for them. The display of the units is done using Netscape’s Live Connect. Live Connect, as stated in chapter 5, allows Java Applets to write out HTML on-the-fly. The first task for the Applet is to obtain a JSObject that represents the frame in which it (the Applet) appears.

```
JSObject win = JSObject.getWindow(this);
```

Once the JSObject is obtained the Applet can call the eval() method to write out the units and optional units to the screen. The applet is also required to write out references to any applets, which may be required later, to their corresponding frames.

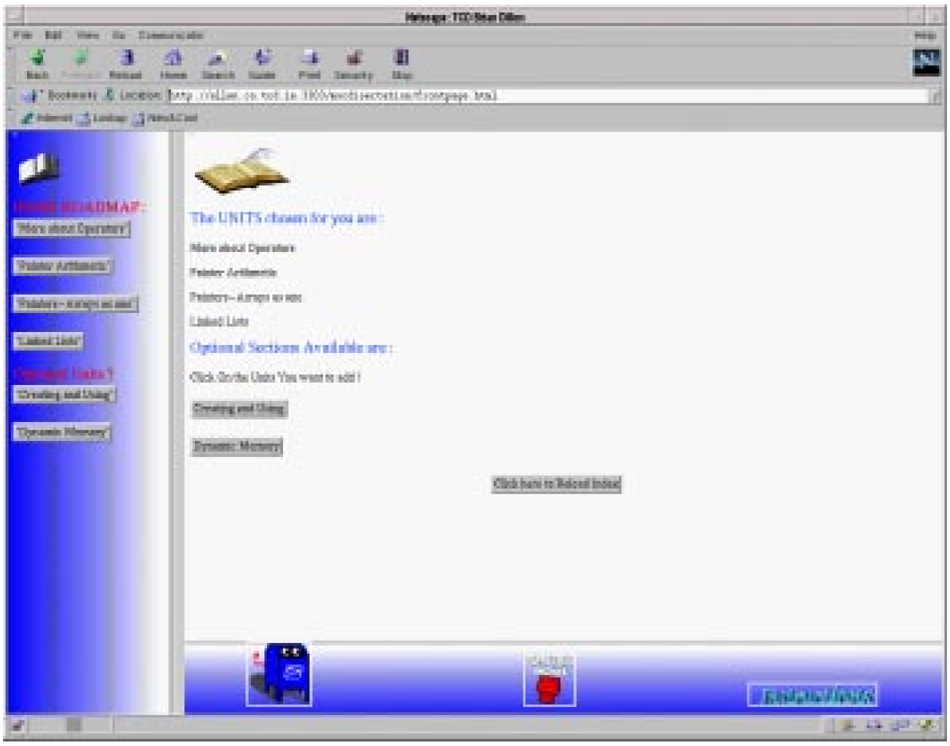
```
win.eval(“document.writeln(‘some string containing the units identifiers and applet
references’)”);
win.eval(“parent.frames[2].document.close()”);//close current document
```

Once the units and the optional units are written out in a list the students are then allowed to select any of the optional units if the so desire.

7.3 COURSE PERSONALISATION

Once the courses are selected for the students, as stated earlier, there may be optional units available for selection. These optional units are stored in the database so that the students are limited in scope of what they can select. If the students were allowed select any units that they wished then they might choose selections which are in no way beneficial to the student's learning, the units may be either too simple or too advanced. This section deals with how this course personalisation is implemented and controlled.

7.3.1 Optional Selections



The UNITS chosen for you are :

More about Operators

Pointer Arithmetic

Pointers– Arrays as one

Linked Lists

Optional Sections Available are :

Click On the Units You want to add !

(Diagram 6.7, Optional selections)

Students firstly click on the buttons corresponding to the optional units which they wish to add to their personal course. Once the button is depressed it calls a procedure, , in the applet, 'updateindex', using Live Connect. The algorithm for this procedure is as follows;

1. Check whether the optional unit has any prerequisites associated with it. The prerequisite dependencies correspond to units which must be completed first by the student. Each unit is allowed one prerequisite to be listed in the 'unit_makeup' table. Multiple prerequisites can be obtained by checking the prerequisites themselves for their own prerequisites. To fulfil this requirement;

    a) Connect to the database if the connection has been closed since the last transaction.
    b) Check if the selection has been made already by the user. This is done by checking the unit_id of the selected optional unit against a list of all the previously selected units.
    c) If the selection has not been already made then;

        Read the dependency from the database. If this indicates that there is a dependency then firstly check to see if the unit_id of the dependent unit has already been selected. If not then add it to the list of selected units and call the recursive function ‘get_dependencies’ to get any additional dependent units.

2. Once it has been verified that the selected optional unit has not been previously chosen and any prerequisite units have been accounted for then the optional units are written out, using ‘win.eval()’, to the bottom of the index.

7.3.2 Sorting the course Index

When the course customising is completed the students may, if they wish, click on the ‘reload index’ button provided. If this button is depressed then the function ‘reload_index’ is called. Firstly the list of optional selected units is sorted. To accomplish this the ‘ordering’ attribute must be retrieved from the ‘section_makeup’ table in the database. The algorithm which is used to sort these units is the ‘Quicksort’ algorithm:

```
Quicksort(int low,  int high,  int ordering[])
{
        int low_val = low;
        int high_val = high;
        //Firstly partition the array into two sections.
                Mid_point = ordering[ (low/high)/2];

        if(low < high) //There has been no crossover
         {
            while(low_val <= high_val)
             {
                // Find the first element that is >= to the partition element
                while( (low_val < high) && (order[low_val] < mid) )
                 {
                   ++low_val;
                 }
                // Find the first element that is <= to the partition element
                while( (high_val  > low) && (order[high_val] > mid) )
                {
                  --high_val;
                }
                //swap,  where indices have not crossed over
                if( low_val <= high_val )
                {
                  temp = ordering[low_val];
                  ordering[low_val] = ordering[high_val];
                  ordering[high_val] = temp;
                  low_val++;
                  high_val--;
               }
            }

          // Next sort the right and left partitions
          if(low_val < high)
                Quicksort(low_val,high,ordering);
          if(low < high_val)
                Quicksort(low, high_val,ordering);

        }//end if
}
```

Once the selected optional units have been sorted in the correct order they must be replaced correctly in the index road-map. This is done by simply comparing each optional unit with each of the pre-selected units. It needs to be pointed out here that this requires accessing the database to retrieve the ‘ordering’ values of the pre-selected units. When this is completed the index contains with all the units in the correct order. All that is then required is to rewrite this reordered index, using Live Connect, onto the screen.

```
JSObject win = JSObject.getWindow ( this );
win2.eval( “parent.frames[0].document.writeln( ‘<applet codebase = \“ .\” archive =
\ “/jdbc/lib/classes111.zip\” code = \“Jdbcnew.class\” name = \“jdbcmy\" width = 5
height = 2 MAYSCRIPT = yes > </applet><p><BODY BACKGROUND =
\“poster.gif\” ><IMG SRC = \“indexbook.gif\” ><p><form><FONT COLOR =
\“#FF0000\”><FONT SIZE = +2>INDEX ROADMAP :</FONT></FONT><form>
“+output_reloaded+” </form><p><p>’ )”);
```

7.4 COURSE MATERIAL ON-THE-FLY

So far it has been seen how students enter and choose their own personalised courses. The next problem which needs to be addressed is how the course material, contained within the database and indicated by the students’ road-map, can be displayed on the screen ?

The students can click on any one of the buttons listed in the course road-map. Once the button is clicked it calls a function, ‘myexecute’, within the applet, ‘Jdbcnew’, which is responsible for retrieving the information from the database. The steps involved in writing out the course material on-the-fly are;

1. Make a connection to the database if it is required.

```
if (conn == null)
        {
          // Load the JDBC driver
          Class.forName (driver_class);
          conn = DriverManager.getConnection(
“jdbc:oracle:thin:@allen.cs.tcd.ie:1521:spgrad” , “bjdillon”,  “bjdillonoracle”);
        }
```

2. Once the connection has been established then the ‘pagelet’ identifiers, of the pagelets which make the unit in concern, must be requested from the database.

```
ResultSet rset2 = stmt2.executeQuery (" select pagelet_id from unit_makeup where
section_id = ‘“+section_id+”’ AND unit_id = ‘“+unit_id+”’”);
```

3. Each ‘pagelet_id’ returned from the database is processed and then a new query to the database is made to retrieve each ‘pagelet’ in turn.

```
while(rset2.next())
  {
        args2= new String(rset2.getString(1));//This gives the pagelet_ids
                                                    //represented as a string
        Integer temp=new Integer(0); //This is the temp integer used for casting

        try
          {
        //We need to do casting here to get from a string->Integer->int for pagelet_id
                temp = Integer.valueOf(args2);
                System.out.println("temp ="+temp+"\n");
            }catch (Exception e){}

          pagelet_id = temp.intValue(); //pagelet_id is the pagelet_id
          System.out.println("pagelet_id ="+pagelet_id+"\n");

          //Now that we have the pagelet_id we need to retrieve the pagelet_info
         ResultSet rset3 = stmt3.executeQuery("SELECT pagelet_info  FROM
pagelet_information WHERE pagelet_id ="+pagelet_id);
          //Here we do the query to get the pagelet_info from the pagelet_information
table
          while(rset3.next())
            {
                //This is where we process the result set containing the pagelet_info
                args3= new String(rset3.getString(1));//the pagelet information

                //Next each pagelet_information is appended to
                // a variable called args4

            }//end processing pagelet info
  }
```

4. Once each ‘pagelet_information’ data is retrieved from the database and appended together in the correct order they must be printed out.

```
win.eval("parent.frames[2].document.writeln(’<p>"+args4+"<p>’)");
```

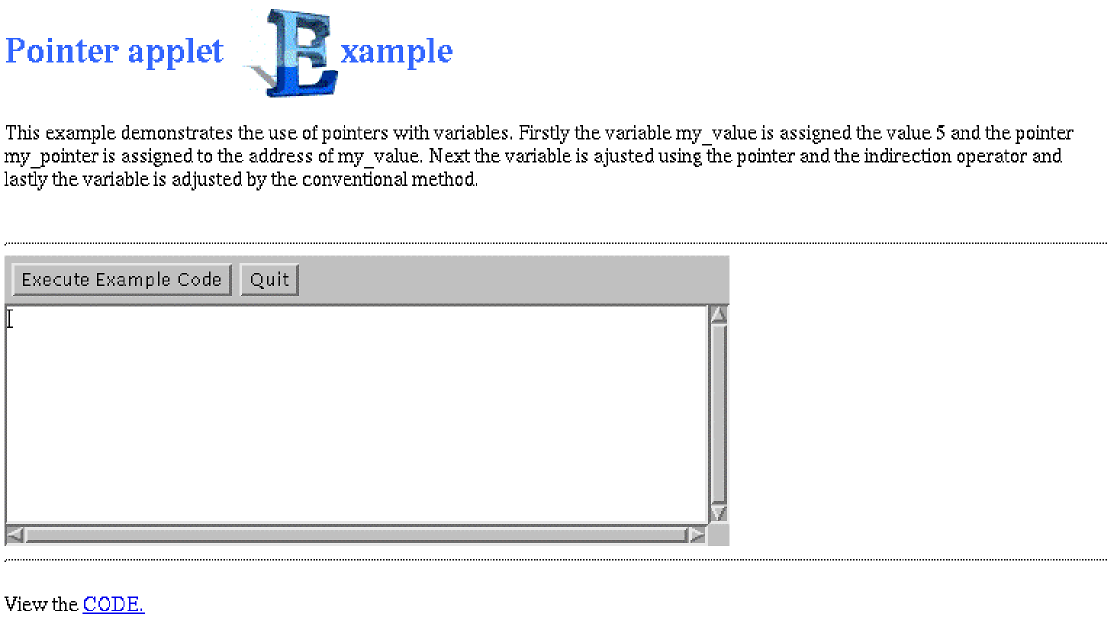
7.5 OTHER ISSUES

This next section will cover specific course related issues such as displaying the C++ code within an applet and how students are allowed to develop SQL[11] on-line during the course.

7.5.1 Student Testing

Student testing was carried out using forms which are Java Script processed. The forms used in the course written to teach C++ pointers are processed in batch form (i.e. all the questions are corrected once the students click the submit button). The questions are corrected and then the results are written out on-the-fly so that the students can view the results. At this point the student can either choose to view the correct answers or try to answer the questions again.

7.5.2 Example C++ Code


Pointer applet Example

This example demonstrates the use of pointers with variables. Firstly the variable my_value is assigned the value 5 and the pointer my_pointer is assigned to the address of my_value. Next the variable is ajusted using the pointer and the indirection operator and lastly the variable is adjusted by the conventional method.

View the CODE.

(Diagram 6.8, Example Applet used to Simulate C++ execution)

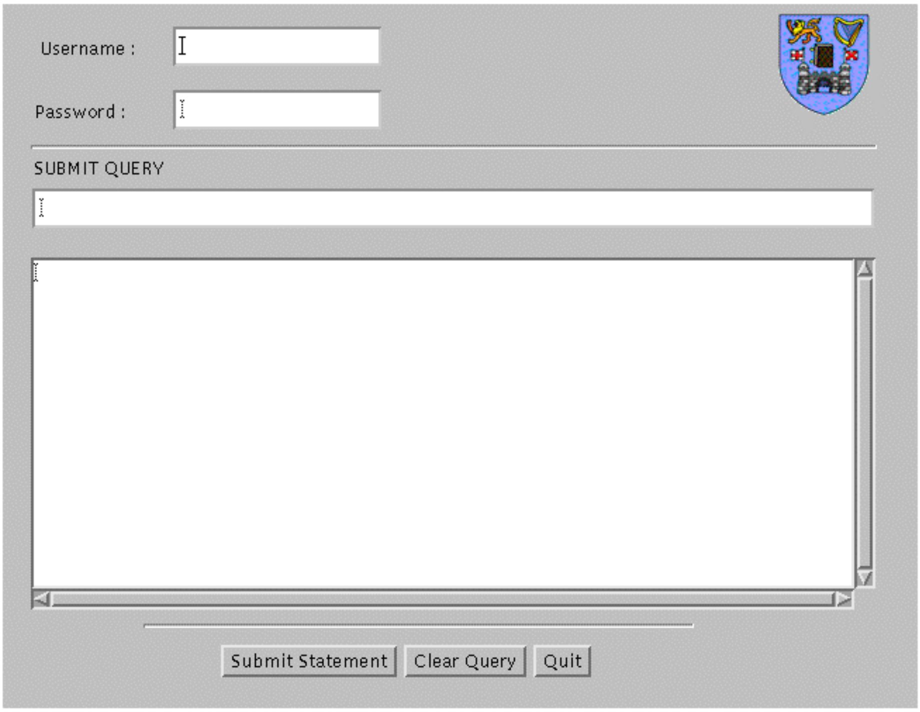
Throughout the course on C++ pointers there are examples such as that seen above where the student are provided with the ability to view code in execution. This adds to the visual and personal feel of the course. Since this implementation of ‘Courseware on-the-fly’ is not provided with a C++ compiler, a method of simulating the C++ code execution was required. The solution used was to write Java applets which when executed produced the same output that would have been produced if C++ programme had been called. The student will not

[11] SQL is the subject of the second course section which is on offer in this example of ‘Courseware on-the-fly’.

know the difference between the actual C++ code provided and the Java programmes simulation of the code.

7.5.3 SQL Development

In the second example course offered there is a tool supplied to the students whereby they can execute SQL statements upon the database.



(Diagram 6.9, SQL development tool)

The students ‘logon’ with their ‘username’ and ‘password’ of their own database accounts. They are then allowed to type in SQL statements to the database. These statements can be of any type, such as;

- Create table.
- Drop table.
- Select from table.

The tools is written as a Java Applet and uses the JDBC-thin driver to connect to the database from the Browser. JDBC drivers use two different types of statement to talk to the database. The first is a query statement for executing queries such as selects.

```
Statement stmt = conn.createStatement ();
ResultSet rset = stmt.executeQuery (student_query);
```

The second type of query statement is used for inserts, updates and creates;

```
PreparedStatement insert_stmt = conn.prepareStatement(student_query);
insert_stmt.executeUpdate();
```

To distinguish between these two types the student query must be first parsed to determine what type of query is being sent. If the statement is a query (i.e. select …) then the query is also parsed to determine the names and numbers of the columns required from the query. Once the query is executed, the results are displayed on the screen.

7.6 CONCLUDING REMARKS

This chapter has dealt with the implementation issues which arose during the development of ‘Courseware on-the-fly’ and the example course material offered. The next chapter will deal with the evaluation of ‘Courseware on-the-fly’ as a teaching aid.

8. CRITICAL EVALUATION

There are several possible objectives in carrying out an evaluation of learning experiences and educational courseware [WAD98]:

- As a basis for administrative decisions such as staffing or promotion.
- For the provision of informative feedback to improve the standard of teaching.
- To provide students with information on which direction they should take in their studies.
- To measure the effectiveness of a teaching aid or the teaching itself. This is an important criteria for pedagogical research.

This chapter will strive to provide formative evaluation on 'Courseware on-the-fly'. As mentioned earlier the main objective of this thesis was to provide 'just in time' courseware composition while adhering to pedagogical rules. The evaluation comprised an evaluation from students participating in the course and the authors own evaluation based on various pedagogical considerations. The first section covers the student evaluation which was performed. The next area to be discussed is the degree of manageability and control afforded to the course director. The final section will then discuss how the various pedagogic issues raised in previous chapters were addressed.

8.1 STUDENT EVALUATION

During the evaluation of 'Courseware on-the-fly' a number of students used the course for studying C++. At the end of their study these students completed a questionnaire[12] to gauge how useful 'Courseware on-the-fly' was for their study. There were a number of different areas covered by the questionnaire, which included;

- The degree of teaching effectiveness. This covered;

  - Whether the students felt that they gained anything from completing the course.
  - How useful the tutorial questions were in aiding self testing.

- Student opinion regarding the automatic composition of course selections and the optional units provided. This section of the questionnaire was used to help judge how effective 'just in time' courseware composition is.

- Whether the method of course presentation (i.e language, colour and graphics) was appropriate. This area also included how the use of simulation and examples aided the students understanding of the course material.

[12] See the appendix for the form used.

The evaluation was paper based and five postgraduate students participated in the evaluation. Unfortunately the evaluation was based on the student's own feelings and there was no examination from the course director.

8.1.1 Teaching Effectiveness

Most of the students felt that they benefited from completing the course. The students were asked if they would be able to complete a "short programme using Linked Lists in C++". Any of those students who before taking the course were unable to do so rectified their problems upon course completion. The tutorial questions provided in the course were designed to facilitate students own self testing. There was unanimous agreement among the students that the optimum number of tutorial questions were asked and that the option to retry questions or view the solutions was beneficial.

8.1.2 Student Expectation

The students were asked how they classified their expertise before taking the course and their expectations upon completion of the course. Student expectation was then gauged using a number of questions;

- Did the selection you chose give the units you required ? Eighty per cent of the students answered yes to this question.
- Did you select any of the Optional Units available to you ? Again Eighty per cent of the students answered yes to this question. This may have been more since not all the selections generated for the students offered Optional units (some of the beginner selections do not include optional units).
- Would you have preferred more or less Optional Units ? Sixty per cent of the students thought that the number of Optional units available was adequate while the remainder of the students felt that they may have benefited if more options were given.

The answers to the above questions show that the vast majority of students were offered the course selections that they required and that they benefited from the facility of customising the course through adding optional units to their road map.

8.1.3 Course Presentation

Course presentation was evaluated using a number of different gauges which included;

- How appropriate was the language used within the course material.
- How easy the course material was to read.
- How appropriate the use of colour was.
- Whether there was good use made of graphics.
- Was anything gained through the use of simulation.
- How straightforward was course navigation.
- Were the examples given throughout the course beneficial.

All of the students agreed that course navigation was clear and straightforward. If this was not the case then the course would almost certainly have been of little benefit to students. All of the students thought the use of examples was beneficial while eighty per cent of them believed that they benefited from the simulation of C++ code in execution. The examples and simulation were used to keep student interest as well as to aid better explanation of the course material. When the students were questioned on the use of language within the course and the easy of which the students could read the material, most were happy although some would have preferred if less use was made of 'animated gifs' and blinking text. This corresponded with the fact that although most students again liked the use of colour and graphics in general, some students would have preferred more plain colours. Since not everyone has the same aesthetic preferences, and most students were happy with the use of colour and graphics, this course presentation issue should be left to the course directors own personal taste.

8.1.4 Other Comments

In the questionnaire students were provided with space where they could give written comment as they deemed appropriate for several of the questions. There was also space at the end of the questionnaire where students could add any additional comments they may have had. This section here will discuss some of the comments from the students.

A number of the students indicated that they liked the general feel of the course and that they liked the idea of being able to customise their own course road map. Others indicated that they liked the inclusion of diagrams in examples. Some students would have preferred the option to view the C++ code and it's execution simultaneously. One student thought the idea of asking participating students their degree of expertise and expected goals, so as to provide only the units which were required, was good.

8.2 COURSEWARE MANAGEABILITY

This is one of the most important issues driving 'Courseware on-the-fly'. Many courses have been previously developed with pedagogic requirements in mind but with no concern given to how the course director change parts of the course without having to concern themselves with the remainder of the course.

8.2.1 Course and Section Management

Within 'Courseware on-the-fly' the course director is provided with the ability to change the structure of the course without having to rewrite any of the course material. This can be done by a simple database 'update' statement. Consider the refresher course on C++ pointers for students of intermediate expertise which includes;

Section_id	Subject_matter	Ordering	Unit_id
Ptrs	Inter Refresh	2	More about Operators

Ptrs	Inter Refresh	3	Pointer Arithmetic
Ptrs	Inter Refresh	4	Pointers_Arrays as one
Ptrs	Inter Refresh	6	Linked Lists
Ptrs	Inter Refreshopt	1	Creating and using
Ptrs	Inter Refreshopt	5	Dynamic Memory

(Diagram 7.1, The section_makeup table)

This course section can be changed in three main ways;

- Shortened; Delete a row, therefore removing a unit, from the database table.
- Lengthened; Add a new unit to the section by inserting a new row into the above table.
- Reordered; The course director may choose to offer a different path through the section. This can be accomplished by the course director changing the order in which the units are offered units, i.e. changing the 'Ordering' attribute in the above table.

The course director may also wish to offer new selections while keeping the old selections for other students. This is achieved once the course director inserts a new section/course into the course/section makeup tables in the database. The course material does not need to be duplicated since all of the course material is stored in the database. This allows for better disk space management, since to re-produce the course material for every new selection would be very resource wasteful/consuming. The course director is also capable of controlling the optional units (e.g. The 'Dynamic memory' and 'Creating and using' units in Diagram 7.1 are optional units in the Intermediate Refresher section) which students may use to customise their own courses. The course director using the same methods as previously explained can extend, reduce or reorder the optional units which students are allowed select.

Section_id	**Unit_id**	**Unit_dependency**	**Pagelet_order**	**Pagelet_id**
Ptrs	Linked Lists	Dynamic Memory	1	Ptr141
Ptrs	Linked Lists	Dynamic Memory	2	Ptr142

(Diagram 7.2, The Unit_makeup table)

The above diagram illustrates what is contained in the unit_makeup table of the database. The 'unit_dependency' attribute is used to prevent students from selecting units without having completed the necessary prerequisites. The course director again has control over what units are considered to be dependent on each other.

There were however a number of drawbacks with the course and section management;

- Course directors must have knowledge of the SQL language since all management is database based. This is an unrealistic expectation since 'Courseware on-the-fly' is supposed to be applicable to non computer science related courses.
- Course directors may accidentally corrupt the course material if they make incorrect changes in the database. This corruption could result in misleading information and/or student frustration.

Both of these drawbacks could be solved with the introduction of a tool which course directors could use to create and maintain courses.

8.2.2 Unit Management

The course director may for some reason need the facility to change the structure or content of a course unit. The last thing that the course director wants to have to do is to rewrite the entire unit. The 'unit_makeup' table is designed in such a way that the course director can change parts of a unit without affecting the rest of the units. It may also be necessary that a unit offered to students with a more advanced level of competence should not be offered to students with weaker skills. To do this the course director could have two units, 'A' and 'a', which cover the same material. Unit 'A' is for the more advanced students so it contains pagelets 1 to 5. Unit 'a' however, is for weaker students so the course director may decide to exclude pagelets three and four so as not to confuse the students. These pagelets or other pagelets may need to be included latter for the weaker students.

Pagelet_id	**Topic**	**Difficulty**	**Dependency**

(Diagram 7.3, The pagelet_meta table)

The facility to control the construction of units at a given time for particular students is very beneficial in this age of mass education. Nowadays at third level, and even to a lesser degree at second level, it is very hard for weaker students to get the extra tutelage they require. If course directors are allowed to construct units which are designed specifically to give better problem explanation for weaker students while not boring the brighter students then both the students and course directors benefit. This can be achieved by adding or subtracting pagelets from units as required. To prevent the course director from making mistakes with the construction of units the 'pagelet_meta' table is provided. This table (Diagram 7.3) allows the course director to enter information about pagelet dependencies, difficulty and the topic covered by each pagelet. If used correctly it enables the course director to construct meaningful and useful course units.

A drawback which is associated with customising course units/courses for students becomes apparent with considering how a group of students could be managed. In a university environment it may be necessary that all the students are tutored to the same level of expertise. An example of this would be a group of students covering material which is examinable upon completion of the course. Another drawback is the simple student profiles used in this example of courseware on the fly. The student profiles which were designed here may not adequately represent all students wishing to participate in the course.

8.3 PEDAGOGICAL REQUIREMENTS

In chapter two a number of educational theories were introduced and discussed. 'Courseware on-the-fly' was developed to gain any benefits available from the discussion. For the purpose of 'Courseware on-the-fly', the constructivist theory was chosen and adhered to.

Constructivists view students as taking an active part in their own education, i.e. they actively interpret what they see according to their individual beliefs. Ausbel claimed that new information must be incorporated into existing cognitive structures in order for it to acquire meaning [TUC92]. If the correct cognitive structures have not been laid in place then new information cannot be added to the students' existing knowledge. Educational aids must then take into consideration students' prior learning and expertise. Constructivist teaching is the provision of a collaborative environment where students have both the opportunity and the means to gain understanding. This section will analyse this implementation of 'Courseware on-the-fly' and see how it strives to provide this environment.

8.3.1 Prior Knowledge

As mentioned earlier it is extremely important that the students prior learning is taken into account in order for a successful course to be offered. This is achieved in two instances in 'Courseware on-the-fly'. Firstly students are required to complete a questionnaire before being offered a course. This purpose of this questionnaire is to gain information about the student so that an accurate student model can be constructed and thus an adequate course road map chosen. The second instance of prior knowledge being considered is in the database where course directors are allowed the facility of placing prerequisites on certain course units (how this is done was explained earlier). These prerequisites prevent students from choosing course units which could hinder their acquisition of knowledge by confusing them.

8.3.2 Student Control

In order for students to gain knowledge they must be actively engaged in the education process. One method of facilitating the students participation is by allowing them some degree of control over what they are taught. In the implementation detailed earlier this is done by allowing the students to select some optional units available. This allows for course customisation which means that students are required to reflect on their own education and not just sit idly back and expect to be 'served' with the information which they require. The course customisation is, however, controlled to some degree by the course director, who is allowed limit the selection which the students make, therefore helping solve the problem of students getting lost in the course or suffering from information drowning. It is very important that the students are guided in the attainment of the knowledge if the knowledge being taught is expected to be useful.

Another less obvious way of facilitating student control was gained through the use of questions and simulations. Students are required to complete tests which are then corrected on-the-fly. These tests can be retaken until the students are satisfied with how they fared. The simulations allow the students to view code in execution. The proverb 'a picture paints a thousand words' is apt here. It is easier for students to learn and understand when they can 'see' the application of the knowledge. This becomes even more apparent in the SQL section, which was added to show that 'Courseware on-the-fly' is not limited to offering one course at a time. Within the SQL section a student tool was developed to allow students to access their own database accounts and try out the SQL which they have learned.

Other forms of student control include the ability of students to choose when and where they want to complete the course, and the electronic mail facility for students to contact each other and the course director. All of these facilitates of student control serve to stimulate the students' exploration and reflection which is necessary for knowledge construction.

8.3.3 Collaboration and Motivation

Collaboration and motivation are some of the central characteristics widely accepted about constructivist education. It refers to the useful interactions between students and teacher. Collaboration is achieved in 'Courseware on-the-fly' through the use of e-mail. It could also be achieved in conjunction with the courses offered if students were given group project work based on the information contained with the courses. Student motivation can come in a number of different forms. The most beneficial type of motivation comes however from the students themselves through interest in what is being taught. This type of motivation is referred to as intrinsic motivation. It is achieved in 'Courseware on-the-fly' through the use of a range of sensory experiences incorporating interactivity and multimedia. Collaboration would have been further improved with the addition of a Web Chat facility.

One such example is the medium of the Web which is used to present the courses. There has in the past few years been a huge growth in the area of the World Wide Web due to ever increasing interest. Most third level students come into contact with the Web at some stage in their university days, so they tend to be interested in what is 'out there' on the Web. Offering courses that are Web based has many advantages including student interest which should not be underestimated. 'Courseware on-the-fly' uses images and diagrams to promote student interest as well as to aid better problem explanation. It should be noted here that a lot of the burden of promoting student interest falls on lecturers. It is they who write the course material, and structure the guidance road maps, bearing in mind the range of students who may choose to study the course. 'Courseware on-the-fly' is a teaching aid and therefore only makes it easier for lecturers and course directors to set up 'interesting' courses for their students.

8.3.4 Other Requirements

A number of other pedagogical requirements must be satisfied for 'Courseware on-the-fly' to be a successful teaching aid;

- The Students should be afforded the opportunity to experiment with the knowledge they have learned. Computers provide the environment required to support this experimentation, within a risk free environment, through the use of simulation.
- The development of a course for 'Courseware on-the-fly' requires support and co-operation from faculty and administration personnel to make in successful. This is a requirement which can only be supported by individual faculties, on both a technical and admistrative level, wishing to use an environment such as 'Courseware on-the-fly'.
- The course material developed for use with 'Courseware on-the-fly', and indeed any course, must be integrated within explicit curricula and clear objectives, content description, teaching method, assessment and course evaluation.

- The failure of many technological educational systems can be attributed, wholly or in part, to human interaction with the computer. If the user interface is designed well it enables students to interact with the course material without complex intermediaries and aids understanding of the knowledge domain and structure [WAD98].

8.4 CONCLUDING REMARKS

This chapter has given a critical analysis of 'Courseware on-the-fly' according to various educational objectives. These educational objectives are of concern to students, course directors and lecturers alike. When all of the three previous sections (The student evaluation, courseware manageability and pedagogical requirements) are taken as a whole it can be seen how well 'Courseware on-the-fly' achieved it's goals. The next chapter in concerned with the achievements of 'Courseware on-the-fly' and any possible future developments and improvements.

9. CONCLUSIONS

This chapter presents conclusions which have been drawn throughout the research, design and implementation involved with ‘Courseware on-the-fly’. It also outlines possible future developments and research into the area of ‘just in time’ courseware composition.

9.1 OVERVIEW OF RESEARCH

‘Courseware on-the-fly’ was developed in support of ‘just in time’ composition of courseware based on an individuals objectives. This required detailed research into the field of epistemology (the theory of knowledge) so as to gain good pedagogic insights. It involved considering the three main pedagogic theories ;

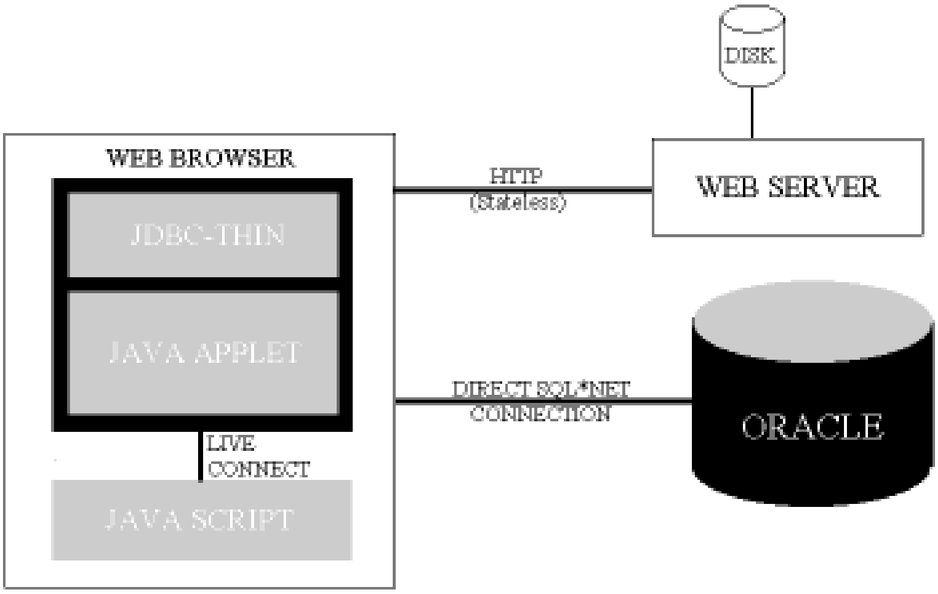
- Behaviourism; This approach to education seeks to find connections between stimulus and response or response and reinforcement, therefore placing great emphasis on the environment. With behaviourism the learner is seen as being a passive part of the learning process, having knowledge ‘dumped in’ [TUC92].
- Cognitivism: This approach not only focuses on the subject’s environment but also on the subjects’ ability to interpret and make sense of something. The student, here, is viewed as an information processing organism to be programmed from the teachers knowledge.
- Constructivism: This approach views knowledge as being individually interpretable and learning as an adaptive process. If this theory is to be adhered to, then students must be allowed freedom to learn as they wish while still gaining the guidance required to enhance the learning activity.

These three theories were examined in-depth so that the best educational aid could be developed. The constructivist theory was the theory most adhered to during the design and implementation of the system. Other insights such as Bloom’s taxonomy were also researched [BLO56], [KRA64]. Bloom’s taxonomy incorporates a set of general and specific categories encompassing all expected outcomes from a lecture or class room situation.

The development of a system to support the pedagogic insights gained from the above research involved considering many different technologies and techniques which included;

- Storage of the Course material.
- Using the Web as the medium for offering ‘just in time’ courses.
- Connecting the Web to the Database.

9.2 REVIEW OF SYSTEM



(Diagram 8.1, The Courseware on-the-fly System)

A system has been developed, tested and evaluated which supports 'just in time' composition of course material based on individual student's objectives. The system is easy to use for both student and course director while adhering to various pedagogic concerns.

Students can initialise a course by completing a brief questionnaire which is used to construct a student profile model. This student model is used when choosing which course to offer the students as well as which optional selections the students are allowed. Choosing course selections for students involves accessing the database for the necessary information stored by the course director. These selections are written out on-the-fly to the screen so that the student can view the units chosen for them. The students may at this stage decide to do one of three things;

- Continue with the selection made for them.
- Select some of the optional units available to them.
- Start over and make a completely new selection.

If the students decide to select some of the optional units they have the option to reload the course index so that the new units selected appear in the correct order (this is done by clicking on the reload index button). Once the students are satisfied with their selection they can view the various course material by clicking the appropriate buttons. Once the button is de-pressed, it causes the course database to be accessed and the material to be retrieved and written out on-the-fly.

While viewing the course material students have the option to complete tutorial questions, watch simulations of code or e-mail their course director. The course material is rich in images and diagrams designed to hold the students interest and improve the clarity of the material being taught.

9.3 SYSTEM IMPROVEMENTS

During the course of the evaluation of ‘Courseware on-the-fly’ a number of possible improvements were identified. The first improvement is the possibility of having student logins. If these were provided students would not be required to complete the student profiling form every time they wishes to study the course material. If this were to be facilitated a table would be required in the database to store the students’ ids and information about the students’ personalised road maps. The table required would have a form similar to the following;

Student_Id	Unit_Id	Unit_Order	Last_Unit

(Diagram 8.2)

- Student_Id; The students login id required so that students can keep their personal course selections.
- Unit_Id; This attribute would contain the unit identifiers so that the course units could be retrieved for the students. This contains both the units selected for the students and their own optional unit selections.
- Unit_Order; The Unit_Order attribute would contain the information on the order in which the course units should be completed.
- Last_Unit; It would be of benefit to students if upon re-entry to the course they are shown the last unit which they previously viewed.

Students could be asked at the start of the course whether they have a course they wish to resume or not. If so then they would be presented with the units contained within their personalised course once they had entered the correct login information. If student passwords are required they could be stored encrypted either in the database or in a file. If students did not wish to resume a course they could enter their login information and then chose a new course selection as they did previously (i.e. complete the student modelling form and choose the optional units required). This time however the students’ course details would not only be written out on-the-fly to the screen but also recorded in the database.

Other course improvements were found to facilitate better teaching practice and / or student interest. These improvements included;

- Breaking down the units into pages to reduce the amount of scrolling that a student needs to do. If this was implemented a student could click on a unit and view the first page in the unit. Once the page was complete, clicking on a ‘next page’ button would cause the next page from the unit to be loaded. Reducing the amount of scrolling that a student needs to do to read the course material allows the student to concentrate more effectively on the information being presented.

- The Introduction of a Web Chat facility to improve student collaboration. Student collaboration is one of the two central characteristics of the constructivist theory on education. A Web Chat facility would provide a means whereby students could communicate with each other, and perhaps the lecturer, in real time. The Web Chat would be of most benefit if the course was provided to students on distance learning courses.

The Student Evaluation performed could have been improved if more participating students were available. Unfortunately as the evaluation was performed before the start of a new college term the only students available to evaluate the system were postgraduates from the computer science faculty. It would have been much more informative if undergraduate students had been given the opportunity to use the course and provide feedback.

9.4 FUTURE WORK

The system developed here could be further developed if the improvements outlined in the previous sections were developed. Other possible development in the area of tele-education would involve a World Wide Web 'spider' which would search the Web for relevant courses based on the information in the databases where the relevant courses are stored. This tool would examine courses, such as those offered using the system developed here, and would produce a list for students based on their indicated requirements. For such a system to be developed communication and understanding between the various educational institutions wishing to take part would be required.

The area of Virtual Reality (computer vision) and Artificial Intelligence should be expected to play a much greater part in education. Education tools developed using some of the insights and research from these areas may help to improve student interest and collaboration. With the cost of computers continually decreasing, more and more people will gain the ability to access computer assisted teaching aids. The increase in computer processing power should help the development of more efficient and effective teaching aids. In Ireland there is a vision that all second level schools will have access to the Web and the learning it provides. The schools are to be provided with Internet connections and a central support service (The National Centre for Technology in Education, NCTE) to provide specialised information services [IRI97].

9.5 CONCLUSIONS

This thesis has provided detailed analysis and design of an implementation of 'just in time' composition of educational material based on student profiles. It firstly examined the different pedagogical considerations and learning theories. This is imperative for the success of any educational course. The various technological concerns were considered so as to develop the system architecture upon which to design and offer course material. Chapter four discussed the issues involved in designing and producing course material for 'Courseware on-the-fly'. The issues discussed here are almost identical to those encountered while developing course material for traditional teaching. Chapter five and six gave the details encompassed in the design and implementation of 'Courseware on-the-fly'. The penultimate chapter gave a critical and quantitative evaluation of the system developed and showed that ‘Courseware on-the-fly’ accomplished it’s objectives.

A. STUDENT EVALUATION QUESTIONNAIRE

Teaching Effectiveness

Answer Before taking the Course

Q1. What would you consider your level of expertise with C++ ?

0 - Beginner	1 - Intermediate	2 - Expert

Q2. Would you be able to write a short programme using Linked Lists in C++ ?

0 - With a lot more study	1 - With a little more study	2 - Yes

Answer After taking the course

Q 3. Would you be able to write a short programme using Linked Lists in C++ ?

0 - With a lot more study	1 - With a little more study	2 - Yes

Q4. Did you feel that you required more tutorial questions ?

0 = Need Much More	1 = Need A few More	2 = The Correct amount of Questions were given.

Q5. Did you feel that you required less tutorial questions ?

0 = Need Much Less	1 = Need A few Less	2 = The Correct amount of Questions were given.

Expectations

Q6. Which course selection did you follow ?

Q7. Did the selection you chose give you the units you required ?

YES	NO

Comments………………

Q8. Did you select any of the Optional Units available to you ?

YES	NO

Comments………………

Q9. Would you have preferred more or less Optional Units ?

More Units	Less Units	The number of Optional Units was adequate.

Comments………………

Course Presentation

Q10. Was the Language Appropriate ?

YES	NO

Comments …………

Q11. Was the material easy to read ?

YES	NO

Comments …………

Q12. Was the use of colour appropriate ?

YES	NO

Comments …………

Q13. Was the use of graphics appropriate ?

YES	NO

Comments …………

Q14. Was the use of simulation appropriate ?

YES	NO

Comments …………

Q15. Was course navigation straightforward ?

YES	NO

Comments …………

Q16. Was there an adequate amount of examples to hold your interest ?

YES	NO

Comments …………

BIBLIOGRAPHY:

[ASH98]
Ashenfelter, John Paul. *WEB DATABASES: How and Why to Use Them in Education.* ED-MEDIA (June 1998).

[BAC96]
Bacher, Ch. *Tools and Services for Authoring on the Fly*. Publication, (1996).

[BLO56]
Bloom, B. S. et al. *Taxonomy of Educational Objectives. Handbook I: The Cognitive domain.* Longmans Green (1956).

[BRU60]
Bruner, J. *The Process of Education.* Harvard University Press (1960).

[CAR96]
Carroll, Erin et al. *Database Programming with JDBC.* Web Techniques (October 1996).

[CEN]
Web Reference; Centra Software Inc - *http://www.centra.com*

[DAL96]
Dalitz Wolfgang et al. *Hyper Wave*. Dpunkt (1996).

[DEI94]
Deitel, H. M. and Deitel P. J. *C++ How To Program*. Prentice Hall (1994).

[DILL95]
Dillenbourg, P. et al. *Collaborative learning and the Internet.* Computer-Aided Instruction Conference (1995).

[DEN96]
Denning, Peter J. *Business Designs For The New University*. ACM Computing Surveys (December 1996).

[ELM94]
Elmasri, Ramez et al. *Fundamentals of Database Systems.* Addison-Wesley (1994).

[EHM]
Ehmayer, Gerald et al. *Connecting Databases to the Web: A taxonomy of Gateways.*

[FLA97]
Flanagan David. Java Script, *The Definitive Guide.* O'Reilly & Associates, inc. (1997).

[FON93]
Fontana, David. *Psychology for Teachers.* British Psychology Society, (1993).

[GAR]
Gardarin, George. *Multimedia Federated Databases on Intranets: Web-Enabling IRO-DB.*

[GEA97]
Geary, David M. et al. *Graphic JAVA.* The SUNSOFT Press (1997).

[GOO98]
Goodman, Danny. *The Java Script Apostle.* Netscape Communications Corporation (1998).

[GREY87]
Grey, G. *Making CBT happen*. Weingarten Publications, Boston (1987).

[GUP98]
Gupta, Abhay. *Implementing JAVA Computing.* IEEE (MARCH 1998).

[HUG67]
Hughes, A.G. et al. *Learning and Teaching*. Longmans, (1967).

[HIL95]
Hiltz, S. R. Teaching in a Virtual Classroom". International Conference on Computer Assisted Instruction, (March 1995).

[IBR95]
Ibrahim, Bertrand, et al. *Advanced Educational Uses of the World Wide Web*. Journal - Computer Networks and ISDN Systems (27), (1995).

[IRI97]
Sundry articles from the *Irish Times* Newspaper (1997)

[KNO67]
Knox, H.M. *Introduction to Educational Method*. Oldbourne Modern Education Library, (1967).

[KRA64]
Krathwohl, D. R. et al. *Taxonomy of Educational Objectives. Handbook II: The Affective domain.* David McKay (1964).

[MAR69]
Maritain, Jacques. *Education at the Cross-roads*. Yale University Press, (1969).

[MAU97]
Maurer, *Hermann. Necessary Ingredients of Integrated Network Based Learning Environments*. Publication, (1997).

[MCC96]

McClanahan, David. *ORACLE Developer's Guide.* Oracle Press (1996).

[NEW96]
Newby, Timothy J. *Instructional Technology For Teaching And Learning*. Prentice Hall (1996).

[OES71]
Oeser, Prof. Oscar. *Teacher, Pupil and Task*. Social Science Paperbacks, (1971).

[PID71]
Pidgeon, Douglas, et al. *An Introduction to Educational Measurement.* Routledge & Kegan Paul, (1971).

[PEB]
Web Reference. Pebblesoft Learning Environment -
http://www.pebblesoft.com/content/Home.html

[POP72]
Popkin, Richard H. et al. *Philosophy Made Simple*. Butler & Tanner, (1972).

[ROS98]
Rosenberg, Dave. *Bringing JAVA to the Enterprise.* IEEE (MARCH 1998).

[RUS67]
Rusk, Robert R. et al. *Doctrines of the Great Educators*. Macmillan, (1967).

[SIP98]
Siple, Matthew. *The Complete Guide to JAVA Database Programming.* McGraw-Hill (1998).

[SME97]
Smeaton, Alan F. et al. *Virtual Lectures for Undergraduate Teaching*. Publication, (1997).

[STA96]
Stauffer, Karen. *Student Modeling and Web-based Learning Systems.* Athabasca University press (1996).

[STO71]
Stones, E. *Educational Psychology*. Methuen & Co. (1971).

[STE96]
Steele, E.D. et al. *Assessing the Learning of Distant Students: Competency Based Instruction.* EDMEDIA (1996).

[TAY71]
Taylor, L.C. *Resources for Learning*. Penguin Books, (1971).

[TCL98]
Sundry papers From a Top Class presentation given by Web Based Training Systems. (1998).

[TUC92]
Tuckey, Catherine. *Uses of New Technology in Higher Education.* Publication, (1992).

[WLH]
Sundry Authors. *World Lecture Hall*. University Of Texas.

[WAD97]
Wade, Vincent, et al. *Network Based Delivery and Automated Management of Virtual University Courses*. Publication, (1997).

[WAD98]
Wade, Vincent, et al. *Evaluating the Design and Deliver of WWW based Educational Environments and Courseware.* EdMedia, (1998).

[WHI98]
Whitehurst, Alan R. Utilising the Student Model in Distance Learning. ACM (1998).

[VAN88]
VanLehn, Kurt. *Student Modelling, Foundations of Intelligent Tutoring Systems.* (1988).

[VCL]
Web Reference. Virtual Classroom - http://eies.njit.edu/

[VER71]
Vernon, M.D. *The Psychology of Perception*. Pelican Books, (1971).

CONTENTS

APPENDIX