GRANULAR APPROACH TO ADAPTIVITY IN PROBLEM-BASED LEARNING

Sally Hong

Abstract

Constructivist approach to learning has been around for quite some time. The constructivist theory has resulted in the development of a wide variety of learning environments, however the problem-based learning (PBL) environment is one of the most ideal and most popular area that implements the constructivism theory. PBL is an attractive approach to foster learner's critical problem solving and self-directed learning skills. However, it is difficult to implement effective PBL environments. A majority of existing PBL environments suffers from the fact that the students easily get inundated by the fine granularity of the problems and loose focus of overall aims of the learning process.

This project has introduced student adaptivity technology into PBL environments to improve the effectiveness and efficiency of the learning process. To demonstrate the idea of PBL with student adaptivity, a web-based prototype is implemented in Process Costing, within the field of Accounting. Based on the architecture of the web-based intelligent educational systems, the problem base module is introduced.

The basic architecture of the system is a typical three-tier, client-server structure. The client tier has the presentation interfaces that are implemented as HTML frames and run in a web browser. The application programs for performing adaptation, which were developed using PHP, reside in the middle layer, and communicate directly with the backend database: problem base, knowledge base that is the third tier. The web server as the communication channel also resides in the middle tier.

With the system, students work on the real world costing calculation problems, and the system evaluates students' performance results on the problems to provide adaptation to the students.

In summary, this project has successfully introduced the student adaptivity into the PBL environment. The strategies used in this thesis can be applied into the pure PBL educational systems to improve their adaptation capability.

Acknowledgements

I would like to thank my supervisor, Dr. Kinshuk for all the help and guidance given to me during this project. I am especially thankful to Dr. Kinshuk for his patience and thorough feedback throughout the whole research and the report writing.

Thanks my husband and mother for supporting me as much as they could.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Introduction

The Problem-Based Learning (PBL), as an effective learning environment for improving students' problem-solving and self-directed learning skills, has been applied some disciplines successfully. But the potentials of PBL are not exploited yet, because its limitations, such that students easily loose focus and get frustrated by lack of adequate guidance. This thesis intends to introduce student adaptivity technologies into PBL environment that aims to exploit its potentials.

This chapter first briefly discusses the basic concepts and the relationship of constructivism and problem-based learning environment. Then the main advantages of problem-based learning and its deficiencies are described, followed by the basic concepts of student adaptivity.

Finally, the chapter outlines the objectives of the project, and introduce the steps of this research and the organisation of this thesis.

1.2 Constructivism Theory and Problem-based Learning

Constructivism is not a totally new approach to learning. In recent years, however, constructivism has become a "hot" issue as it has begun to receive increased attention in a number of different disciplines, including instructional design.

Like most other learning theories, constructivism has multiple roots in the philosophical and psychological viewpoints of this century. A number of contemporary cognitive theorists have adopted the constructivism, which considers:

knowledge is a function of how the individual creates meaning from his or her own experiences during learning and understanding. In general, constructivism is a learning theory about how humans learn and know knowledge. This theory is characterized by the three following propositions (Savory and Duffy, 1995):

A) Knowledge is in human's interactions with the environment: this is the core concept of constructivism. What humans understand is a process of the content, the context, the activity of the learner and the learner's goals.

B) Cognitive conflict is the stimulus for learning and determines the organization and nature of what is learned: when human being are in a learning environment, there is some stimulus or goal for learning.

C) Understanding is influenced by the processes associated with collaborative learning: the experiential world includes, most importantly, the social environment.

The features of constructivism outlined above have founded a wide variety of learning environments (Duffy et al., 1993), and the problem-based learning environment is the almost ideal and most popular area that implements the constructivism theory (Savory and Duffy, 1995):

The PBL model has its roots in the apprenticeship, or learning-by-doing, method. It emphasizes a "real-world" approach to learning: a student-centered process that is both constructive and collaborative. PBL is a motivating way to learn, as learners are involved in active learning, working with real problems. PBL intends to foster learner's problem-solving and self-directed learning skills. The PBL provides a more stimulating environment for the learner and a more enjoyable environment for students and teachers.

However, in practice, PBL is difficult to implement, with or without computer-based support. In the traditional face-to-face PBL, teachers must be specially trained as guides and students often become frustrated by the lack of information. In the computer assisted intelligent PBL environment, since the PBL does not limit what students may choose to learn, and the process may provide little guidance on the best ways of achieving learning goals, students may be concerned that their learning strategies are misdirected or inefficient. Thus, it is much harder for student who learn through the computer-based intelligent learning systems with PBL, and students may easily loose focus during the learning process and become frustrated by feeling out of control in their study.

1.3 Student Adaptivity in Computer-based Intelligent Learning Systems

Student adaptivity in intelligent learning systems provides the systems ability to adapt themselves to the goals and tasks of students by monitoring their performance. The adaptivity is one of core components in intelligent learning environments. The application of student adaptivity can create better learning environments in intelligent learning systems.

The main reasons that student adaptivity is so important in intelligent learning systems are as follows:

- A wide student spectrum: the student spectrum may be from one extreme (naïve) to another extreme (advanced), that means that students may have very different backgrounds, learning styles, individual preferences, and knowledge levels. The systems with student adaptivity are able to improve the effectiveness and efficiency of learning.

- The intelligent systems focus more on student centred learning. Theses systems are usually are used by the students of different places and in different contexts. The student adaptivity in these systems concerns about individual student's preference and knowledge level, with aim to make the learning more efficient and effective.

1.4 Objectives of the Research

PBL is an attractive approach to foster learner's problem-solving and self-directed learning skills, and these skills are crucial in general life and career. However, PBL is difficult to implement, with or without computer-based support. One of major reason is that students easily loose focus and get frustrated by lack of adequate guidance and help available in PBL environments. This project attempts to address this problem by introducing student adaptivity technologies into computer-based PBL learning environment.

The main objectives of the project are as follows:

- Investigate PBL and student adaptivity technologies, and design a framework of student adaptivity function in computer-based PBL learning environment.

- Design and implement a web-based PBL prototype to test and evaluate the student adaptivity in PBL environment.

1.5 The Steps in This Research

The following steps summarize the plan of this project that is used to reach the objectives described above:

- *Step 1:* Literature review: it covers constructivism, problem-based learning, student adaptivity in intelligent learning systems.

- *Step 2:* Design the system architecture: it introduces student adaptivity mechanism into PBL learning environment, and build up an architecture based on the literature review.

- *Step 3:* Exploit some related technologies, such as CGI, Client-server, and so forth.

- *Step 4:* Implement the prototype.

The outline of steps is shown in Figure 1.1.

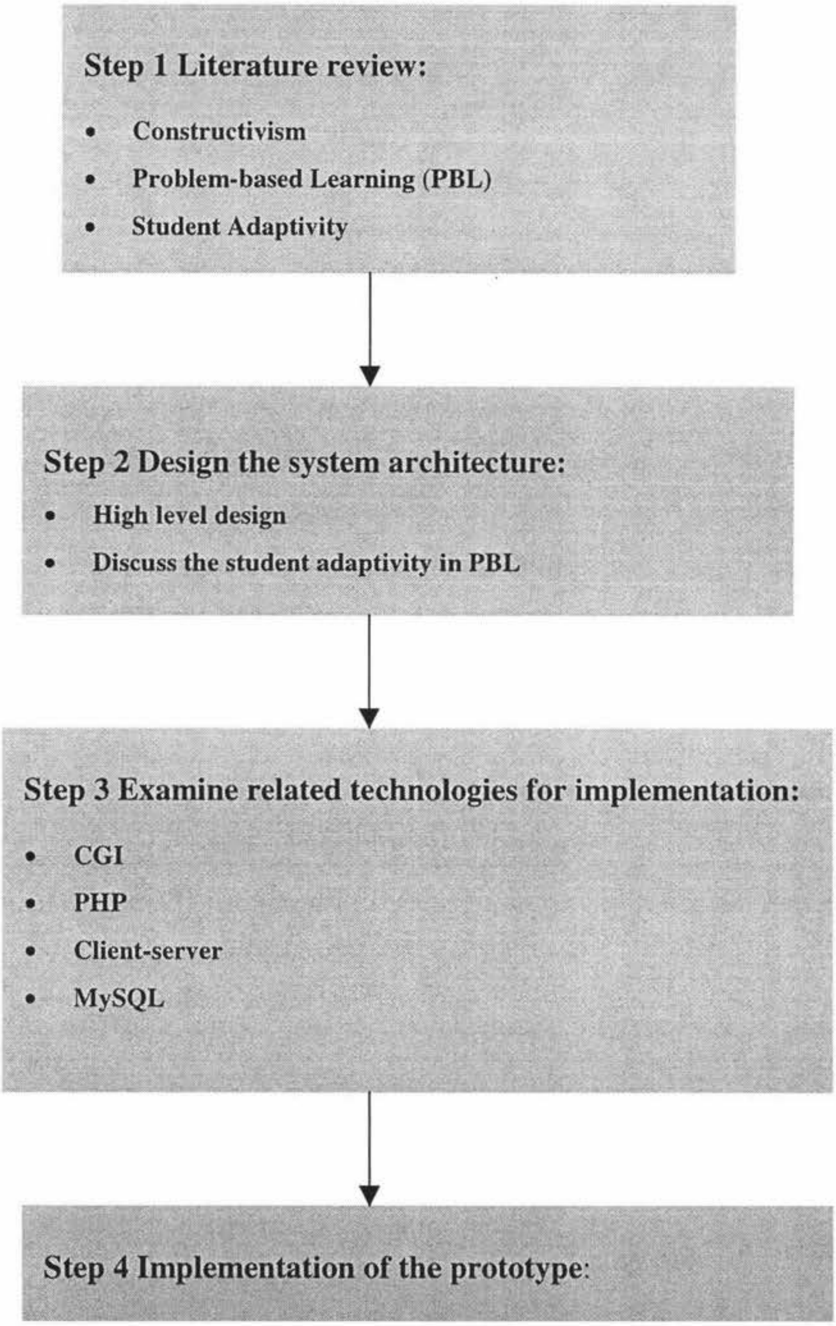

Figure 1.1: The research steps of this project

1.6 Organization of the Thesis

The thesis consists of 8 chapters. They are outlined below:

Chapter 2 reviews the history and features of the constructivism theory.

Chapter 3 moves from constructivism onto problem-based learning area. It overviews the features of PBL, and discusses its advantages and disadvantages.

Chapter 4 investigates student adaptivity technology and introduces this technology to address the problem with PBL in theory.

Chapter 5 designs the system architecture based on the above research results, and describes its mechanism.

Chapter 6 examines the related technologies, such as CGI, PHP, Client-server, and describes the implementation of the prototype based on the architecture.

Chapter 7 discusses how the student adaptivity works in this prototype.

Chapter 8 concludes the contributions of this project and presents some further consideration about this project in future.

Chapter 2

Literature Review: Overview of Constructivism

2.1 Introduction

In this chapter, the history of learning theories is briefly reviewed, and the importance of learning theories is discussed. Then the definition of constructivism and its features are described in detail.

2.2 Historical Foundations

Current learning theories have roots that extend far into the past. The problems that today's theorists and researchers investigate are not new but simply variations on a timeless theme:

- Where does knowledge come from?
- How do people come to know?

Two opposing positions on the origins of knowledge-empiricism and rationalism have existed for centuries and are still evident, to varying degrees, in the learning theories of today. A brief description of these views will help to understand the current learning viewpoints of behaviorism, cognitivism, and constructivism.

- *Empiricism* is the view that experience is the primary source of knowledge (Schunk, 1991). That is, organisms are born with basically no knowledge and anything learned is gained through interactions and associations with the environment, and knowledge is derived from sensory impressions. Those impressions, when associated contiguously in time and/or space, can be hooked together to form complex ideas. For example, the complex idea of a tree, as illustrated by Hulse et al. (1980), can be built from the less complex ideas of branches and leaves, which in turn are built from the ideas of wood and fiber, which are built from basic sensations such as greenness, woody odor, and so forth. Based on this theory, instructional design issues focus on how to manipulate the environment in order to improve and ensure the occurrence of proper associations.

- *Rationalism* is the view that knowledge derives from reason without the aid of the senses (Schunk, 1991), and humans learn by recalling or "discovering" what already exists in the mind. For example, the direct experience with a tree during one's lifetime simply serves to reveal that which is already in the mind. The "real" nature of the tree becomes known, not through the experience, but through a reflection on one's idea about the given instance of a tree. In summary, knowledge arises through the mind. Based on this theory, instructional design issues focus on how best to structure new information in order to facilitate the learners' encoding of this new information, as well as the recalling of that which is already known.

The empiricist provided the framework for many learning theories during the first half of this century, and it was against this background that behaviorism became the leading psychological viewpoint (Schunk, 1991). Because behaviorism was dominant when instructional theory was initiated (around 1950), the instructional design technology that arose alongside it was naturally influenced by many of its basic assumptions and characteristics.

In the late 1950's, learning theory began to make a shift away from the use of behavioral models to an approach that relied on learning theories and models from the cognitive sciences. Psychologists and educators began to de-emphasize a concern with observable behavior and stressed instead more complex cognitive processes such as thinking, problem solving, language, concept formation and information processing (Snelbecker, 1983). Within the past two decàde, a number of authors in the field of instructional design have openly and consciously rejected many of traditional behavioristic assumptions in favor of a new set of psychological assumptions about learning drawn from the cognitive sciences. Whether viewed as an open revolution or simply a gradual evolutionary process, there seems to be the general acknowledgment that cognitive theory has moved to the forefront of current learning theories (Bednar et al., 1991). This shift from a behavioral orientation (where the emphasis is on promoting a student's overt performance by the manipulation of stimulus material) to a cognitive orientation (where the emphasis is on promoting mental processing) has created a similar shift from procedures for manipulating the materials to be presented by an instructional system to procedures for directing student processing and interaction with the instructional design system (Merrill et al., 1981).

The philosophical assumptions underlying both the behavioral and cognitive theories are primarily objectivistic; that is: the world is real, external to the learner. A number of contemporary cognitive theorists have begun to question this basic objectivistic assumption and are starting to adopt a more constructive approach to learning and understanding: knowledge is a function of how the individual creates meaning from his or her own experiences. Constructivism is not a totally new approach to learning. Like most other learning theories, constructivism has multiple roots in the philosophical and psychological viewpoints of this century, specifically in the works of Piaget, Bruner, and Goodman (Perkins, 1991). In last decade, however, constructivism has become a "hot" issue as it has begun to receive increased attention in a number of different disciplines, including instructional design (Bednar et al., 1991).

2.3 Importance of Learning Theory

Learning theories typically are divided into two categories: behavioral and cognitive. However, in the last decade constructivism is added as a third category because of its recent emphasis in the instructional design literature (e.g., Bednar et al., 1991; Duffy and Jonassen, 1991; Winn, 1991). In many ways the viewpoints of three categories overlap, yet they are distinctive enough to be treated as separate approaches to understand and describe learning.

Why these learning theories are highlighted in research and practice?

- First, learning theories are the source of verified instructional strategies, tactics, and techniques. Knowledge of a variety of such strategies is critical when attempting to select an effective prescription for overcoming a given instructional problem.

- Second, learning theories provide the foundation for intelligent and reasoned strategy *selection*. Instructional designers must have an adequate repertoire of strategies available, *and* possess the knowledge of when and why to employ each. This knowledge depends on the designer's ability to match the demands of the task with an instructional strategy that helps the learner.

- Third, *integration* of the selected strategy within the instructional context is of critical importance. Learning theories and research often provide information about relationships among instructional components and the design of instruction, indicating how specific techniques/strategies might best fit within a given context and with specific learners (Keller, 1979).

- Finally, the ultimate role of a theory is to allow for reliable *prediction* (Richey, 1986). Effective solutions to practical instructional problems are often constrained by limited time and resources. It is paramount that those strategies selected and implemented have the highest chance for success. As suggested by Warries (1990), a selection based on strong research is much more reliable than one based on "instructional phenomena."

It is essential to mention further that students exposed to the three instructional approaches described above would gain different competencies, and there is no single "best" approach. Given that learning is a complex, drawn-out process and learning is influenced by many factors, the suitable learning theory it depends on the concrete learning.

2.4 Definition of Constructivism

Constructivism is a theory about learning and is a philosophical view on how we come to understand or kno. Before we go further to discuss the constructive learning theory, let us have a look the definition of learning.

2.4.1 What is Learning?

Learning has been defined in numerous ways by many different theorists, researchers and educational practitioners. Although universal agreement on any single definition is nonexistent, many definitions employ common elements. The following definition by Shuell (1991) incorporates these main ideas: "Learning is an enduring change in behavior, or in the capacity to behave in a given fashion, which results from practice or other forms of experience" (p. 2).

To the objectivists, knowledge and truth exist outside the mind of the individual and are therefore objective, and learners are told about the world and are expected to replicate its content and structure in their thinking (Jonassen, 1991). The role of education in the objectivist view is therefore to help students learn about the real world. It is asserted that there is a particular body of knowledge that needs to be transmitted to a learner. Learning is thus viewed as the acquisition and accumulation of a finite set of skills and facts.

To the constructivist, learning is described as a change in meaning constructed from experience (Newby et al., 1996). Constructivists believe that knowledge and truth are constructed by people and do not exist outside the human mind (Duffy and Jonassen, 1991). This is radically different from what objectivism conceives learning to be.

Constructivism emphasizes the construction of knowledge while objectivism concerns mainly with the object of knowing.

2.4.2 Features of Constructivism

To compare and understand learning theories, Schunk (1991) lists five definitive questions:

1. How does learning occur?
2. Which factors influence learning?
3. What is the role of memory?
4. How does transfer occur? and
5. What types of learning are best explained by the theory?

2.4.2.1 Learning Occurrence in Constructivism

Constructivism is a theory that equates learning with creating meaning from experience (Bednar et al., 1991). Even though constructivism is considered to be a branch of cognitivism (both conceive of learning as a mental activity), it distinguishes itself from traditional cognitive theories in a number of ways. Most cognitive psychologists think of the mind as a reference tool to the real world; constructivists believe that the mind filters input from the world to produce its own unique reality (Jonassen, 1991).

Constructivists do not share with cognitivists and behaviorists the belief that knowledge is mind-independent and can be "mapped" onto a learner. Constructivists do not deny the existence of the real world but contend that what we know of the world stems from our own interpretations of our experiences. Humans *create* meaning as opposed to *acquiring* it. Since there are many possible meanings to glean from any experience, we cannot achieve a predetermined, "correct" meaning. Learners do not transfer knowledge from the external world into their memories; rather they build personal interpretations of the world based on individual experiences and interactions.

Thus, the internal representation of knowledge is constantly open to change; there is not an objective reality that learners strive to know. Knowledge emerges in contexts within which it is relevant. Therefore, in order to understand the learning, which has taken place within an individual, the actual experience must be examined (Bednar et al., 1991).

2.4.2.2 The Factors of Influencing Learning in Constructivism

Both learner and environmental factors are critical to the constructivist, as it is the specific *interaction* between these two variables that creates knowledge.

Constructivists argue that behavior is situationally determined (Jonassen, 1991). Just as the learning of new vocabulary words is enhanced by exposure and subsequent interaction with those words in context (as opposed to learning their meanings from a dictionary), likewise it is essential that content knowledge be embedded in the situation in which it is used. Brown et al (1989) suggest that situations actually co-produce knowledge through activity. Every action is viewed as "an interpretation of the *current* situation based on an entire history of previous interactions" (Clancey, 1986). For this reason, it is critical that learning occurs in realistic settings and that the selected learning tasks be relevant to the students' lived experience.

2.4.2.3 The Role of Memory in Constructivism

The goal of instruction is not to ensure that individuals know particular facts but rather that they elaborate on and interpret information. A concept will continue to evolve with each new use as new situations, negotiations, and activities recast it in a different, more densely textured form. Therefore, "memory" is always under construction as a cumulative history of interactions. Representations of experiences are not formalized or structured into a single piece of declarative knowledge and then stored in the head. The emphasis is not on retrieving intact knowledge structures, but on providing learners with the means to create novel and situation-specific understanding by "assembling" prior knowledge from diverse sources appropriate to the problem at hand. For example, the knowledge of "design" activities has to be used by a practitioner in too many different ways for them all to be anticipated in advance. Constructivists emphasize the flexible use of pre-existing knowledge rather than the recall of prepackaged schemas (Spiro et al., 1991). Mental representations developed through task-engagement are likely to increase the efficiency with which subsequent tasks are performed to the extent that parts of the environment remain the same. Memory is not a context-independent process.

Clearly the focus of constructivism is on creating cognitive tools which reflect the wisdom of the culture in which they are used as well as the insights and experiences of individuals. There is no need for the mere acquisition of fixed, abstract, self-contained concepts or details. To be successful, meaningful, and lasting, learning must include all three of these crucial factors: activity (practice), concept (knowledge), and culture (context) (Brown et al., 1989).

2.4.2.4 Transfer of Knowledge

The constructivist position assumes that transfer can be facilitated by involvement in authentic tasks anchored in meaningful contexts. Since understanding is "indexed" by experience, the authenticity of the experience becomes critical to the individual's ability to use ideas (Brown et al., 1989). An essential concept in the constructivist view is that learning always takes place in a context and that the context forms an link with the knowledge embedded in it (Bednar et al., 1991). Therefore, the goal of instruction is to accurately portray tasks, not to define the structure of learning required to achieve a task. If learning is decontextualized, there is little hope for transfer to occur. One does not learn to use a set of tools simply by following a list of rules. Appropriate and effective use comes from engaging the learner in the actual use of the tools in real-world situations. Thus, the ultimate measure of learning is based on how effective the learner's knowledge structure is in facilitating thinking and performing in the system in which those tools are used.

2.4.2.5 The Best Types of Learning Based on Constructivism

The constructivist view does not accept the assumption that types of learning can be identified independent of the content and the context of learning (Bednar et al., 1991). Constructivists believe that it is impossible to isolate units of information or divide up knowledge domains according to a hierarchical analysis of relationships. Although the emphasis on performance and instruction has proven effective in teaching basic skills in relatively structured knowledge domains, much of what needs to be learned involves advanced knowledge in ill-structured domains.

Jonassen (1991) has described three stages of knowledge acquisition (introductory, advanced, and expert) and argues that constructive learning environments are most effective for the stage of advanced knowledge acquisition, where initial misconceptions and biases acquired during the introductory stage can be discovered, negotiated, and if necessary, modified and/or removed. Jonassen agrees that introductory knowledge acquisition is better supported by more objectivistic approaches (behavioral and/or cognitive) but suggests a transition to constructivistic approaches as learners acquire more knowledge which provides them with the conceptual power needed to deal with complex and ill-structured problems.

In summary, central to the tenet of constructivism is that learning is an active process. In her Educational Psychology textbook, Woolfolk (1993) described the constructivist view of the learning process as follows: "The key idea is that students actively construct their own knowledge: the mind of the student mediates input from the outside world to determine what the student will learn. Learning is active mental work, not passive reception of teaching" (p485). Hence, according to the constructivist perspective, learning is determined by the complex interplay among learners' existing knowledge, the social context, and the problem to be solved. Although various authors have described the characteristics of constructivist instruction (e.g. Brooks and Brooks, 1993; Cognition and Technology Group, 1993; Collins et al., 1991; Honebein et al., 1993), two characteristics are central to these constructivist descriptions of the learning process:

- Good problems: constructivist instruction asks learners to use their knowledge to solve problems that are meaningful and realistically complex. The problems provide the context for the learners to apply their knowledge and to take ownership of their learning. Good problems are required to stimulate the exploration and reflection necessary for knowledge construction.

- Collaboration: the constructivist perspective supports that learners learn through interaction with others. Learners work together as peers, applying their combined knowledge to the solution of the problem. The dialogue that results from this combined effort provides learners with the opportunity to test and refine their understanding in an ongoing process. There is another aspect of collaboration in a constructivist learning environment, which involves the role of the teacher.

2.5 Summary

This chapter has reviewed constructivism theory in detail. Constructivism is not a totally new learning theory, and is developed from multi-disciplines such as the philosophy and psychology. In last decade, constructivism has been highlighted by a number of different disciplines, especially instructional design.

One popular summary of constructivism theory is pointed out by Lebow (1993). He noted that:

> ...traditional educational technology values of replicability, reliability, communication, and control (Heinich, 1984) contrast sharply with the sever primary constructivist values of collaboration, personal autonomy, generativity, reflectivity, active engagement, personal relevance, and pluralism (p.5).

The instructional principles outline by Lebow leads to a wide variety of learning environments. However, the problem-based learning environment is the almost ideal and most popular area that implements the constructivism principles. In chapter 3, the problem-based learning area is going to be reviewed.

Chapter 3

Literature Review: Overview of Problem-Based Learning

3.1 Introduction

This chapter reviews one of the best exemplars - problem-based learning for constructivism, which is a constructivist learning environment and is described by Barrows (1986, 1998) at the Southen Illinois Unvieristy Medical School. The history of problem-based learning and its definition are discussed. Then we look into some existing computer assisted learning environment where problem-based learning has been implemented.

3.2 History

Problem-based Learning (PBL) as a general model was developed in medical education. The original problem-based curriculum at McMaster University, featuring small learning groups with a faculty tutor, was established some thirty years ago (Neufeld, 1974, Spaulding, 1991). As a newly created school, McMaster began with this revolutionary problem-based curriculum after a reasonably luxurious four-year opportunity to set it up. A few years later two more new schools, widely spaced across the globe one at Maastricht University in the Netherlands and the other at University of Newcastle in Australia, undertook problem-based learning curricula. There was much cross-fertilization between all three schools. Many faculties from Maastricht and Newcastle spent months to years at McMaster as they were planning their curricula.

Over twenty years ago, the University of New Mexico School of Medicine created another problem-based learning revolution by establishing an alternative, problem-based curriculum emphasizing rural, primary care. It was designed for a small number of students and ran parallel with their traditional curriculum. The initial development of this curriculum was financed by an external grant from the Kellogg Foundation and required changes in curriculum design and teaching that were quite different from their traditional curriculum. Promoting this new method for teaching students among the faculty was a new and difficult challenge for the small, dedicated, alternative curriculum faculty at New Mexico (Kaufman, 1985). They were pioneers in creating the change from traditional to problem-based curricula. Subsequently, other medical schools such as Harvard, Bowman Gray, Rush and Southern Illinois University established alternative, parallel curricula and faced the challenge of faculty and curriculum conversion.

Over the last decade, many new and developing countries around the world have also initiated problem-based curricula (Kantrowitz, 1987), such as schools including Harvard and Sherbrooke University in Canada have converted their entire, previously conventional, curricula to problem-based learning (Anderson, 1991, Des Marchais et al., 1992). The University of Kentucky School of Medicine developed a problem-based learning curriculum in surgery and more recently problem-based learning has been incorporated in psychiatry and surgical clerkships at Southern Illinois University.

Currently, it has been refined and implemented in over sixty medical schools. At this point problem-based learning cannot be considered as an experimental method in medical education. It has probably been more thoroughly studied and evaluated than have the traditionally accepted educational methods used in medical school.

In recent years, the Problem-based Leaning model has been adopted in an increasing number of other areas including business schools (Milter and Stinson, 1993), Schools of Education (Duffy, 1994); Architecture, Law, Engineering, Social Work (Boud and Feletti 1991); and high school (Barrows and Myers, 1993).

3.3 Definition of PBL

Although a number of researchers have given the definition to PBL from different views, it should be noted that the definition of PBL is ambiguous (Albanese et al., 1993). For the most part, PBL is a learner-centered educational method. Problem-Based Learning involves the use of complex, "real-world" problems as the stimulus and framework for learning. It is based on the premise that students will be motivated to "want to know" and solve the problem posed because it is presented in a context that simulates real world situations. Acquiring knowledge in the context in which it is meant to be used facilitates recall and application of concepts and skills learned (Gijselaers, 1996). In PBL, learners are progressively given more and more responsibility for their own education and become increasingly independent of the teacher for their education. PBL produces independent learners who can continue to learn on their own in life and in their chosen careers. The responsibility of the teacher in PBL is to provide the educational materials and guidance that facilitate learning.

PBL is a motivating way to learn as learners are involved in active learning, working with real problems and what they have to learn in their study is seen as important and relevant to their own lives. The objectives of PBL is to produce learners who will:

- Engage the problems they face in life and career with initiative and enthusiasm.

- Problem-solve effectively using an integrated, flexible and usable knowledge base.

- Employ effective self-directed learning skills to continue learning as a lifetime habit.

- Continuously monitor and assess the adequacy of their knowledge, problem-solving and self-directed learning skills.

3.4 Features of PBL

3.4.1 Basing on Real World Problems

In PBL, the learning is based on the messy, complex problems encountered in the real world as a stimulus for learning and for integrating and organizing learned information in ways that will ensure its recall and application to future problems. The problems in PBL are also designed to challenge learners to develop effective problem-solving and critical thinking skills.

3.4.2 The Curriculum Based on Problems

The PBL curriculum is different from the conventional instruction. The series of problems encountered by learners with the learning process make up the curriculum for PBL.

The related problems are put together as a group to stimulate learning of content appropriate to the course. In the PBL process learners characteristically learn far more and in areas relevant to their personal needs.

3.4.3 Learner Self-directed Learning Process

In the PBL learning process learners encounter a problem and attempt to solve it with information they already possess allowing them to appreciate what they already know. They also identify what they need to learn to better understand the problem and how to resolve it.

Once they have worked with the problem as far as possible and identified what they need to learn, the learners engage in self-directed study to research the information needed finding and using a variety of information resources (books, journals, reports, online information, and a variety of people with appropriate areas of expertise). In this way learning is personalized to the needs and learning styles of the individual.

The learners then return to the problem and apply what they learned to their work with the problem in order to more fully understand and resolve the problem.

After they have finished their problem work the learners assess themselves and each other to develop skills in self-assessment and the constructive assessment of peers. Self-assessment is a skill essential to effective independent learning.

3.4.4 The PBL Teachers Work as Tutors

In problem-based learning, the traditional teacher and student roles change. The students assume increasing responsibility for their learning, giving them more motivation and more feelings of accomplishment, setting the pattern for them to become successful life-long learners.

The faculties in turn become resources, tutors, and evaluators, guiding the students in their problem solving efforts. As learners become more proficient in the PBL learning process the tutor becomes less active.

3.4.5 Emphasizing on Collaborative Learning Group

In a PBL environment, students are asked to solve a given problem. The problem is posed to the students before relevant information has been presented through any medium, including texts or lectures, about the subject matter underlying the problem.

Learning is ideally in small groups of 5 to 7 learners, which students work together to analyze the problem and determine what information they already have and what information they do not know and need to learn in order to solve the problem.

3.5 Advantages and Disadvantages of PBL

3.5.1 Advantages of PBL

The justification for PBL is based on its compatibility with modern theories of adult learning, together with evidence of efficacy in some areas. Most students enjoy the active participation, which PBL fosters and consider the process to be relevant, stimulating and even fun (Des Marchais, 1993). On the other hand, teachers tend to enjoy the increased student contact (Albanese et al., 1993). There is convincing evidence that PBL fosters self-directed learning skills, and this may help students to be life-long learners (Shin, 1993, Blumberg, 1992). The main advantages of PBL are:

- Learning and teaching is more enjoyable for students and teachers
- The learning environment is more stimulating
- Self-directed learning skills are enhanced and retained
- Promotes deeper rather than superficial learning
- Increases interaction between students and faculty
- Improves motivation to learn

3.5.2 Disadvantages of PBL

The main disadvantages of PBL are:

- The criticism most often voiced is that PBL is costly, in demands of staff time and teaching materials and other physical resources. Both initial and on-going costs should be considered -- considerable energy and resources are needed over several years to develop the curriculum and to train tutors and students in the PBL process. Once up and running, a PBL curriculum can be demanding of staff time; Des Marchais estimated that the introduction of PBL at Canada's University of Sherbrooke increased the teaching load by 30% (Des Marchais, 1993).

- PBL can be stressful and discomfort for both students and staff, at least until they become familiar with the process (Berkson, 1993). Most students come to PBL from educational backgrounds where teachers direct learning and are accustomed to subject-based learning passing exams, one subject at a time. They are not used to setting their own learning objectives or creating their own learning plans.

By contrast, PBL does not limit what students may choose to learn, and the process may provide little guidance on the best ways of achieving learning goals. Students may be concerned that their learning strategies are misdirected or inefficient. This is the main concern that this thesis tries to address by introducing student adaptivity technologies to help students.

- Learn less: Learner want to learn it all, immediately, but they can not (Woods, 1994). In fact, for each case, individuals learn a bit, but in depth; over many problems, they learn much, in depth.

- Inefficiency: another disadvantage of PBL is its relative inefficiency (Woods, 1994) -- some research suggests that PBL curricula cover about 80% of what might be accomplished in a conventional curriculum in the same period (Albanese et al., 1993).

3.6 Implementation of PBL in Computer Assisted Learning Environment

Over the last decade, computer-assisted learning technologies have been used in various aspects in problem-based learning to improve PBL's effectiveness and students' study results. Computer technologies are mainly used within PBL environment in the following aspects.

3.6.1 Effective and Efficient Information Resources

In the conventional problem-based learning, students need to research the information and solution for the problems faced by using a variety of information resources, such as books, journals, reports, experts and so on, which spread widely in spatial distance.

Computer occurs as a convenient and effective alternative for information resource. For example, Medici (Devitt et al., 1995), Web browser support for PBL (Pennell et al., 1995).

The Medici consists of a shell constructed in HyperCard and Toolbook into which cases can be placed. The cases are built up as a series of blocks and the information can be explored from one block to the next by sequential or branched.

The Web browser support for PBL system was intended to improve undergraduate biological science students' investigative competencies. It uses HTML to organize the learning support material, and the knowledge base includes text, images, sounds and digital video. The material is organised to allow the solution of a graded series of problems with theory experiments and references (Figure3.1). The Internet links are included to general biochemistry references and in the glossary to link to relevant information. A web search engine is provided to supplement the sources. The system largely applies hypermedia and multimedia technologies to support the PBL. It is a typical example in using computer technologies to provide resources for support PBL.

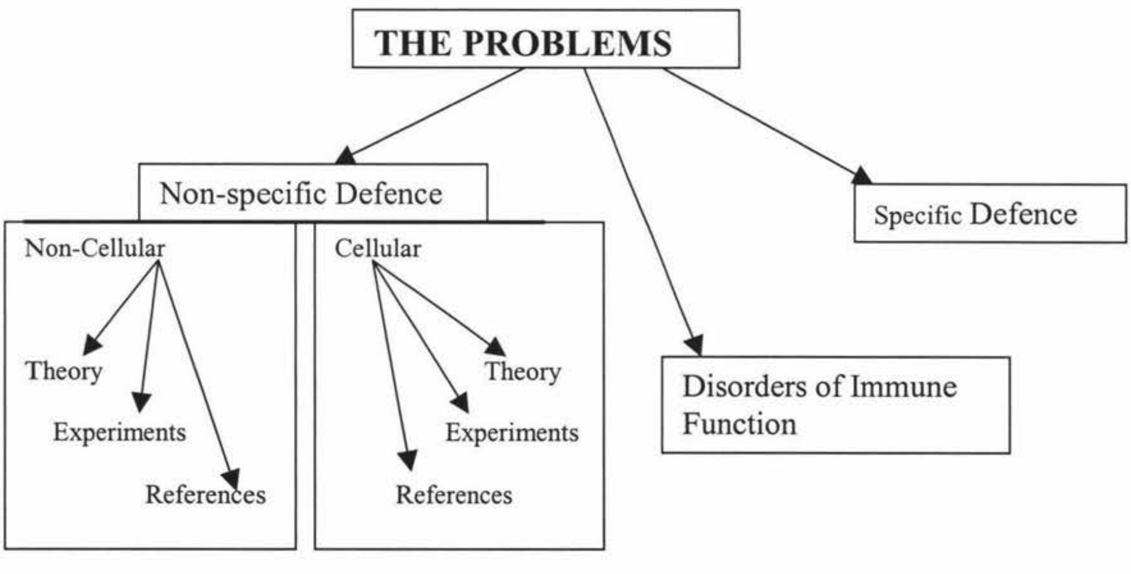

Figure3.1: Relationship between the problems to the knowledge base

3.6.2 Simulation of Real World Problems

One of main characteristics in PBL process is based on real world problems, but in traditional PBL the 'real world' problem most are presented in text and some pictures. Computer programs can be effectively used to simulate real world problems occurring in economics, medicine, and so forth. For example the future cash flow changes of a big company can be simulated to wee how different input values will affect the profit of the company. In medicine, the development of simulation programs that offer clinical simulation with gradual revelation of data can allow the presentation of realistic scenarios that are instructional, stimulation and enjoyable.

3.6.3 Support Collaborative Learning

Various educational theorists, such as Vygotsky (1978), Lave and Wenger (1991), have stressed the importance of social interaction to learning. The term "Computer Support for Collaborative Learning" (CSCL) is used to focus the study of the use of computer-based collaboration tools in instruction. CSCL supports:

- Learning across geographic and temporal distances
- Group, peer-to-peer, and peer-to-expert learning
- Synchronous/asynchronous help
- Formal and informal instruction

In a PBL environment, learning processes depend upon collaborative interaction within the learning group. For instant, Collaborative Learning Laboratory (CLL) (Koschmann, et al., 1990) introduced CSCL technology into PBL environment. Each participant has a private screen and a shared public screen. The projected screen serves the role of a blackboard, flip chart, and overhead projector in a more traditional face-to-face meeting. The private screen serves as an electronic desktop to be organized and used by the individual participant. The researchers claimed that the CLL could enhance the PBL methodology at the different process during learning.

3.6.4 The Advanced Distance Education (ADE) Project (ADE, 2002)

The Advanced Distance Education (ADE) is a project that has been largely exploiting the potential of computer technologies in medical distance education. ADE project is developing tools for the creation of adaptive, Web-based courseware incorporating artificial intelligence. It developed both problem-based learning (PBL) courses and traditional case-based courses. In particular, an agent-assisted, computer-based architecture (Figure3.2) is applying to implement the PBL.

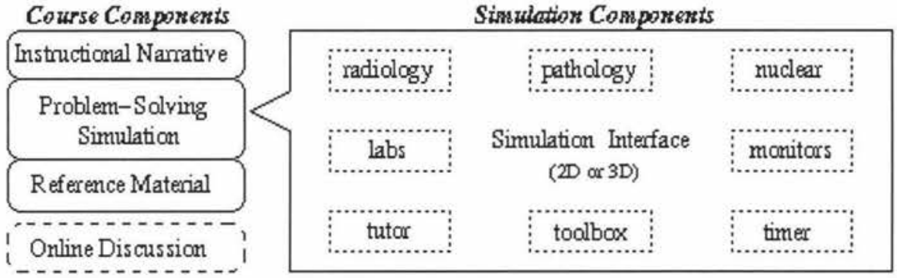

Figure 3.2: Key elements of a typical course module (Johnson and Shaw, 1997)

The PBL in ADE project employs the above three computer assisted techniques that are providing information resources in a variety of formats, problem-solving simulation, and online discussion to support collaborative learning. The students are given an instructional narrative and presented with the case, in the form of an interactive problem solving simulation that includes a pictorial view of the patient and information about the patient such as lab tests, pathology results, and X-rays are accessed by students. On-line reference materials and on-line discussion are provided.

In addition, a pedagogical agent (Adele) monitors the student's actions and provides feedback and guidance as appropriate. The Adele agent is used to adapt the system to the needs of individual students. But the adaptivity in the system is limited. The agent developed is based on the specific patient case and the similar students' background. It defines the task order or pattern to be performed and then compares the student's action against the task order to recommend the actions to perform. It can not provide further helps, such as students need to refer to or further study some relevant material according to students' level (e.g. beginning or advanced), or some further guidance for next step.

3.7 Summary

Following the review of constructive theory, this chapter discussed what is the problem-based learning (PBL) and its feature and advantages/disadvantages.

Finally, the computer assisted PBL are described. The computer technologies are comonly used to provide information resources, simulation of the real world problems and collaborative learning facilities. However, few systems implemented the student adaptivity in PBL environment to address one of difficulties in PBL application – students easily become frustrated by lack of information.

In the next chapter, student adaptivity and its feature are discussed.

Chapter 4

Literature Review: Student Adaptivity

4.1 Introduction

This chapter reviews student adaptivity technology. Some basic concepts and mechanisms of student adaptivity are discussed. Finally, the key technique of student adaptivity in intelligent learning systems: student model is presented.

4.2 Overview of Student Adaptivity

4.2.1 What is Student Adaptivity?

The student adaptivity is a core research topic in intelligent learning environment (Nikov et al., 1999). The student adaptivity enable the systems to tailor themselves based on individual student's performance and background. The goal of student adaptivity is that not only "every one should be computer literate" but also that "computers should be user literate" (Browne et al., 1990). It also makes the learning systems more effective and efficient.

The level of adaptivity within intelligent learning environments varies from system to system. On one hand, systems can be adaptabel – allowing the users to change certain system parameters and adapt systems behavior accordingly. On other hand, systems can be adaptive – changing their behaviour automatically based on the system's assumptions about the user needs.

Oppermann et al. (1997) summarized the complete spectrum of adaptivity that exist in learning systems (Figure 4.1).

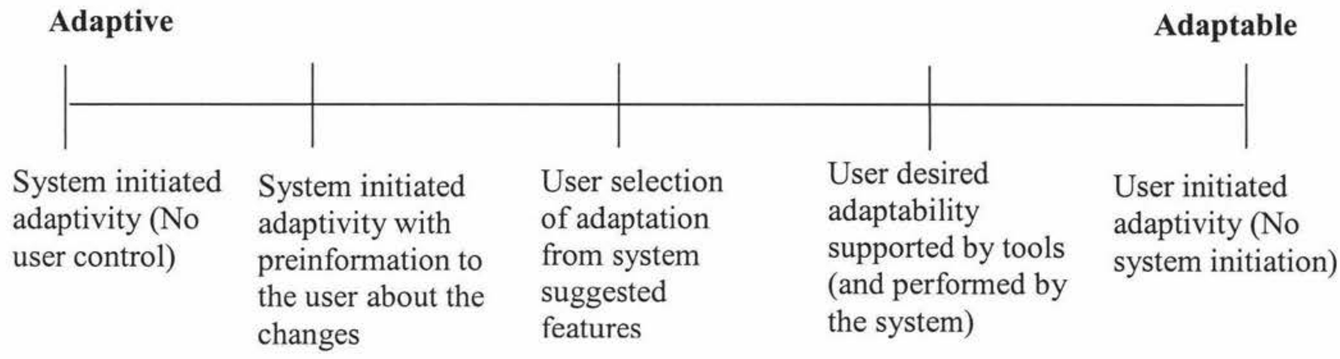

Figure 4.1: the whole spectrum of the concept of adaptation (Oppermann et al., 1999)

4.2.2 Mechanism of Student Adaptivity in Intelligent Educational Systems

Based on the architecture of intelligent educational systems, the tutoring module can be seen as the engine to accomplish the adaptive processes, and student model is the key information for adaptation. Figure 4.2 is the zoom out structure of intelligent educational system (Han, 2001). From figure 4.2, it is clear that the tutoring module is the brain of the system and the window to interact with the students. It captures the student's performance and communicates with student model, then considers the student's personal situation and other related information to draw the suitable adaptivity decisions for the specific student.

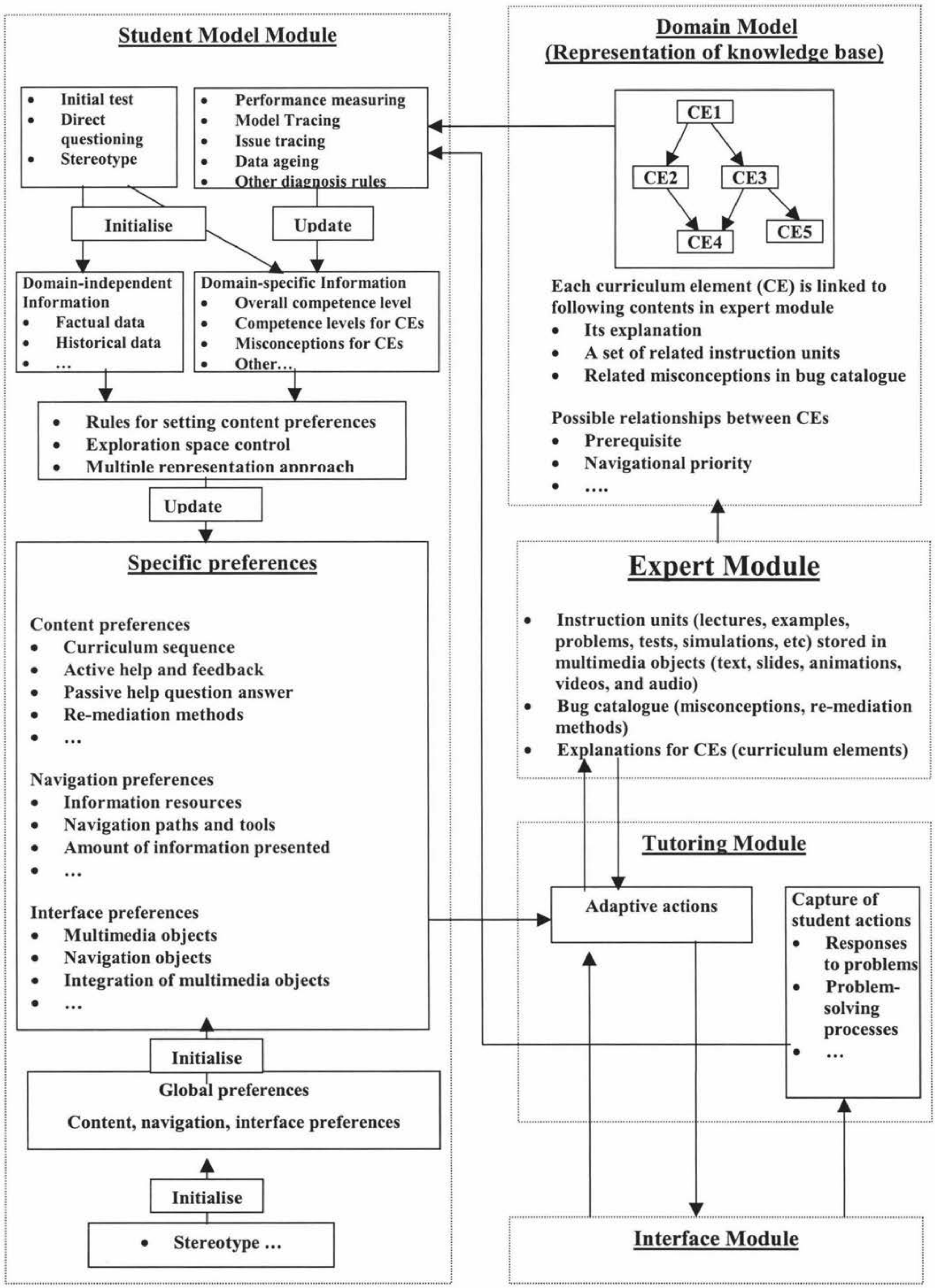

Figure 4.2: Structure of student model module and its relationship with other modules (Han, 2001)

4.2.3 Implementation of Student Adaptivity from Application Level

In an intelligent learning system, the same teaching material can be presented in different formats and order for a specific student. That is *adaptive presentation.* In addition, in web-based environment, the links among pages can be adapted for individual students. That is called as *adaptive navigation.*

- Adaptive presentation: Based on a specific student's learning goals, knowledge level, personal preferences, the content and format of courseware (including text and multimedia), can be adapted to fit the student's needs (Brusilovsky, 2000). The adaptive presentation is a mature technique and has been tested successfully for improving student's learning effectiveness and efficiency (Boyle et al., 1994).

  There are different approaches used to implement the adaptive presentation, such as sorting information order, giving prerequisite explanations, adding additional explanations, comparing related concepts, and preparing variants for one concept.

  Sorting information order: The system is able to order the information of a concept to fit a particular student's needs.

  Giving prerequisite explanations: Before presenting information about a concept, if system recognises that the particular student needs to know the concept's prerequisite information, then the prerequisite information will be presented to the student first.

  Adding additional explanations: To the same concept, different individual student may access information at various levels based on the student's model. Some information may be hidden from a specific student if he does not need to know this information (Höök et al., 1996). This approach was used in KN-AHS (Kobsa et al., 1994), ITEM/IP (Brusilovsky, 1992), and Anatom-Tutor (Beaumont, 1994).

  Comparing related concepts: In some situations, system may decide to compare the related concepts to help students making analogies and hence getting deeper understanding.

Preparing variants of one concept: Depending on studies' capabilities, system may provide the same content or concept in different formats, suitable to particular students' attributes.

There are some existing systems using adaptive presentation technology, for example, ITEM/IP (Brusilovsky, 1992), De Bra's adaptive course on Hypertext (Calvi et al., 1997), ELM-ART, and InterBook (Diego et al., 2000).

- Adaptive navigation: the basic mechanism of adaptive navigation is similar to the adaptive presentation. But the adaptive navigation attempts to adapt and define the navigation paths or navigation space for a specific student. In practice, it is implemented by modifying the visible links, such as sorting, annotating or hiding the links of pages. There are different levels of approaches to provide the navigation adaptivity to the students:

  Global orientation: It helps the students to understand the overall course structure and his/her absolute position in the hyper-space.

  Global guidance: It helps students find the shortest way to the information they want. The most direct method is to provide the student with the links to follow at every browsing step.

  Local orientation: It helps the student by pointing out what is around and his/her relative position in the local hyper-space.

  Local guidance: It helps students move one navigation step by suggesting the most relevant links from the current node. The example systems are: ISIS-Tutor, SYPROS (Gonschorek et al., 1995), HyperTutor and Hynecosum (Vassileva, 1996), WebWatcher (Armstrong et al., 1995), and Adaptive HyperMan (Mathé et al., 1996).

4.3 Student Model

Student adaptivity is the core of intelligent educational system. How successful an intelligent educational system is largely depends on how good student adaptivity it provides. The student model is the most important component that makes individualised adaptation possible.

4.3.1 What is Student Model?

A student model stores the student's skill, knowledge, interests, practices, goals, any other information related to the student (Self, 1994). It is constantly updated in accordance with the dynamic features of the student skills, knowledge, and proficiency during the learning process, because the student is acquiring knowledge continuously (Neilsen, 1990, Brusilovsky, 1994, Vassileva, 1995).

Brusilovsky (1994) pointed out that student models are usually classified according to the nature and form of information contained in the models. Based on the relationship with the subject domain, the information held in student models could be divided into two groups: domain dependent information and domain independent information.

- Domain dependent information: It represents a reflection of the student's state and level of knowledge and skills in term of a particular domain (Brusilovsky, 1994)

- Domain independent information: It contains the student's learning goals, cognitive aptitudes, measures for motivation state, preference about the presentation method, factual and historic data, etc.

4.3.2 Purposes of Student Model

The purposes of student model is to monitor and record the student's behaviour, answers, and problem-solving strategies, and then to analyse them in order to determine the level of the student's understanding of subject at any given moment.

In an intelligent educational system, the tutoring module is the inference engine that performs a wide range of functions, such as knowledge development, error remediation, domain content representation, exploration space control, student assessment, and so forth; and the student model is the base of the tutoring module. For instance, when a student answers a question or completes an exercise, the system will use the student model as a base to determine the best possible instructional path.

4.4 Summary

Intelligent educational systems with student adaptivity have been developed for about three decades. The research has proved that the capability of student adaptivity in systems improves student's learning effectiveness and efficiency.

Based on the investigation above, a prototype was developed to demonstrate how to apply student adaptivity in PBL environment. In the next chapter, system analysis and design of the prototype are discussed in detail.

Chapter 5

System Analysis and Design

5.1 Introduction

This chapter analyses the system, outline the main requirements of the system, and presents the high level architecture of the system and its three-tiers structure. Then the database design is discussed in detail, including database ER diagram, schema, and individual tables' contents.

5.2 Requirements

According to the literature review and the investigating results, this system intends to introduce the student adaptivity into a web based problem-based learning environment. It aims to prevent students loosing the focus during PBL learning processes and avoid the frustration that generally results in typical PBL due to the lack of available information, and limited help. The domain selected for this system is Process Costing, with the field of Accounting. The system attempts to help accounting students understand the Process Costing by studying in the problem-based environment. The main characteristics of this system are:

- *Problem base:* The system uses real world problems designed at different complexity levels. Each problem usually consists of several parts.

- *Flexible assessment for each problem based on student performance:* Each parts of the problem is assessed according to its degree of difficulty by give weights the each part, keeping the total for the whole problem as 100.

- *Teacher defines the criteria for assessing student's level:* Teacher is able to define and modify the criteria for assessing student's performance level for each problem. For example, the teacher can define that if the points student achieves for a problem are less than 30, the level of student in this problem is "beginning".

- *Student adaptivity*: The system is able to adapt based on the student level. For example, if the student is at the beginning level, the system will present the related information from introduction and basic concepts to examples, and let the student do the same or similar exercises again.

5.2.1 Requirements of Problem Base

The problem base provides the real world problems to be worked on by the students. The default order of the problems is defined by course designer.

The requirements of problem base are outlined as follows:

- Problems need to be as close to real world problem as possible.
- One problem is a standalone module to be presented or worked on.
- Each part of a problem is marked by difficulty degree that is out of 100 for the problem.
- It is assumed that each part of a problem may be related to none or one concept in the knowledge base.
- It is assumed that each part of a problem may be related to none or one example in the knowledge base.
- It is assumed that each part of a problem must have only one correct solution.
- One problem must be related to only one section, and/or one example in the knowledge base.
- Each problem has its solution.
- Criteria for assessing student competent level is defined by the teacher as a generic criteria. This approach makes it easy to modify the criteria

5.2.2 Requirements of Knowledge Base

Knowledge base contains all domain content. Sections, specific concepts, and examples are the major components of the representation of knowledge base. The relationships of these components form the foundation of the *content tree* and the *concept network.*

The requirements of a section unit are summarised as follows:

- A section can be a content section or non-content section.
- A content section can be learning section or example section.
- A section may be related to none, one, or more sections.
- A learning section may be related to none, one, or more concepts.
- A learning section may be related to none, one, or more example section.

The requirements for an example are as follows:

- Each example must be related to one or more sections.
- Each example may be related to one or more concepts.

The requirements for a concept are described as follows:

- Each concept must be related to only one learning section.
- No concept should be related to a non-content section.

5.2.3 Developing Student Model

This system focuses on providing dynamic student adaptivity while student is undertaking problem-solving tasks. The student model is not the concern of this project, so it was not implemented in this system.

5.3 The System Design

5.3.1 System Conceptual Model

Figure 5.1 shows the high level architecture of this system. Based on the architecture of the web-based intelligent educational systems, the problem base module is introduced into this architecture.

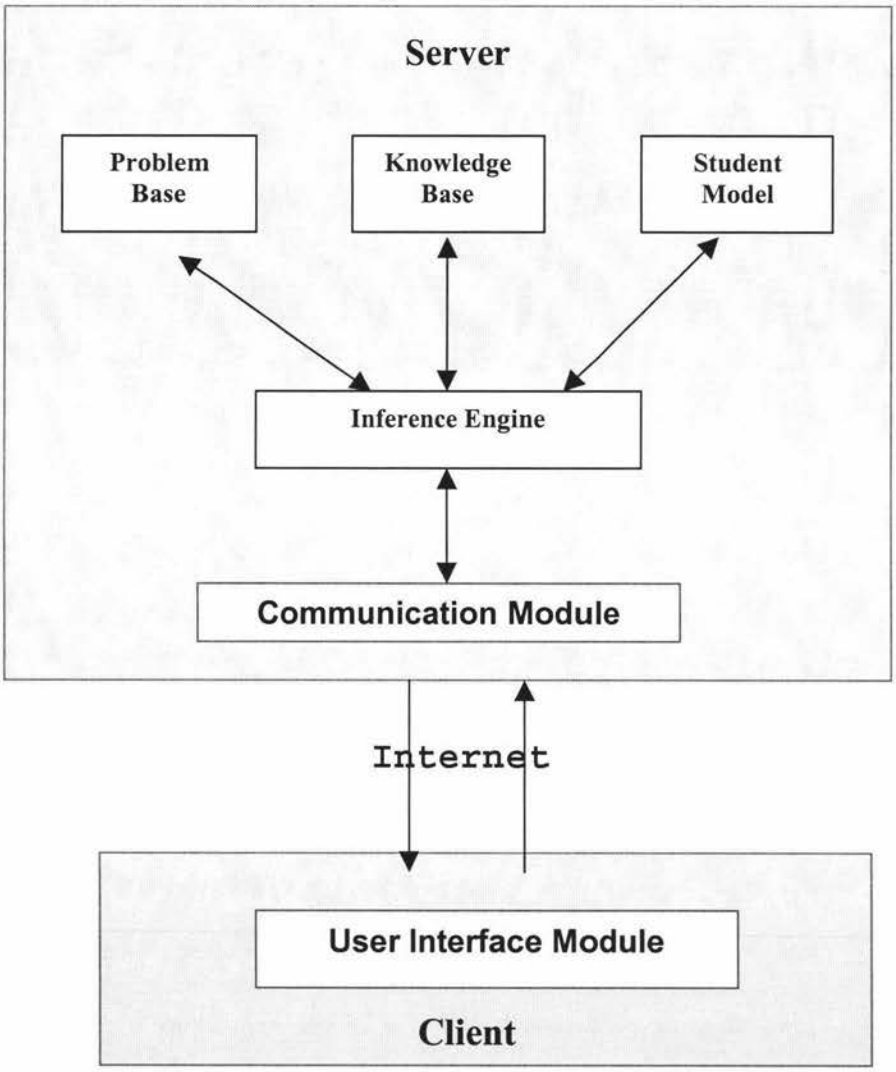

Figure 5.1: The architecture of web-based intelligent educational systems

As shown in the figure 5.1, the basic architecture of the system is a typical three-tier, client-server structure. The client has the presentation interfaces that are implemented as HTML frames and run in a web browser. The application programs for performing adaptation reside in the middle layer, and they communicate directly with the backend database: problem base, knowledge base, and student model. The web server as the communication channel also resides in the middle tier. Figure 5.2 shows the three-tier architecture of the system, and the main functions of the components.

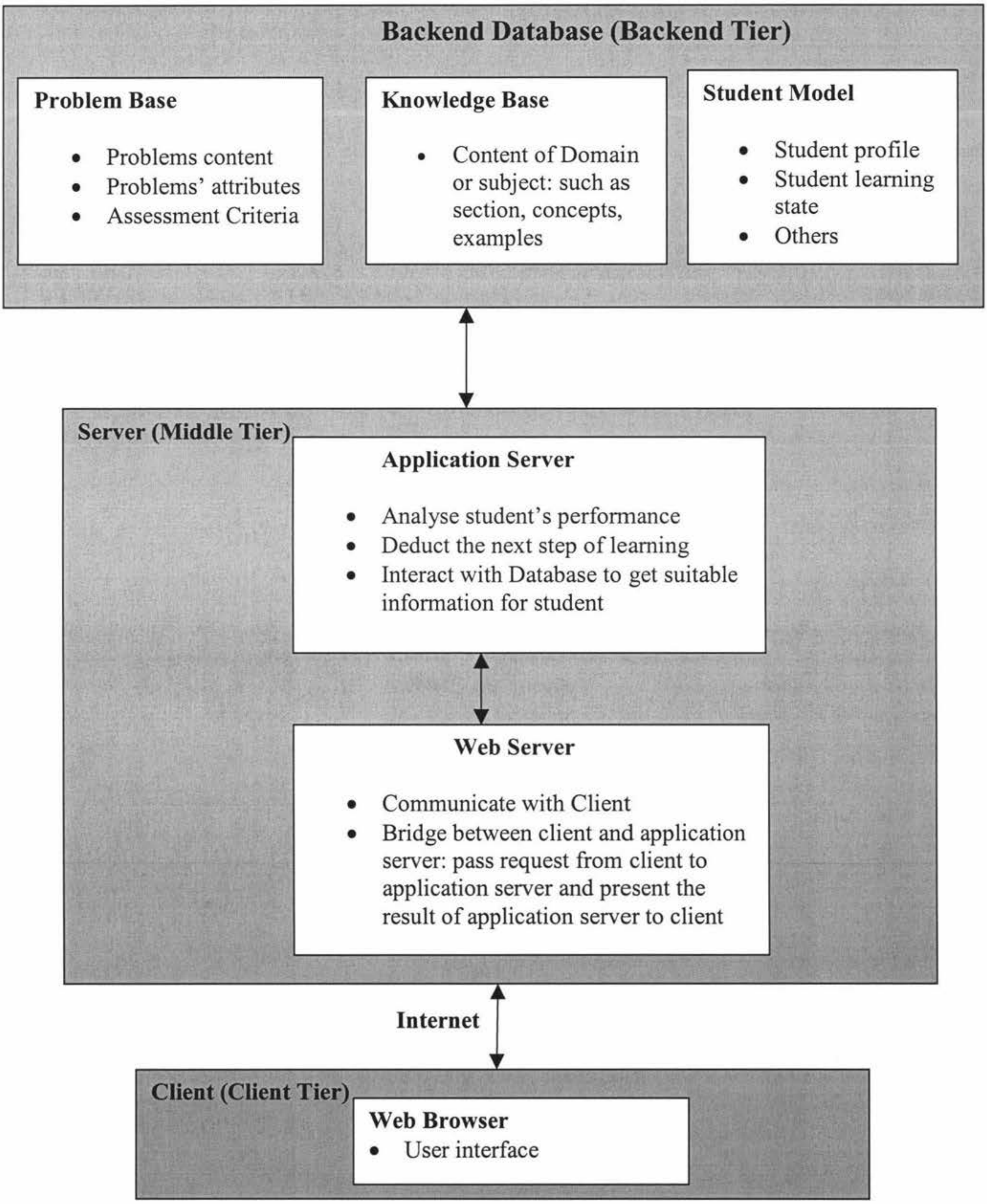

Figure 5.2: The three-tier architecture of the system and the main functions of the components.

- Backend tier: basically it is a database management system (DBMS) that stores and manages the data required by the middle tier's application program. It also resides on the server side, and physically may reside on the same or different machine. In this system, it is implemented in MySQL.

- Middle tier: the middle tier that resides in the server side is the engine of the system. This layer usually consists of web server and application server.

  The web server, we use Apache in this system, communicates with client to get requests from the client and passes the requests to application server if they need to be processed. Then the web server also is responsible for sending the information back to client, normally they are web pages.

  The application server receives client requests transferred by the web server, processes the data contained in the requests. In this system, the application server will analyse the student performance contained in the client requests, and refer to the problem base stored in the backend tier (database) to assess the student's performance, then deduct the next step the student must take. Finally, it will generate a client response based on the deducted results and may refer to knowledge base stored in the backend tier (database), then send it back to the client through web server. The application server also handles the student model initialisation and update processes. In implementation level of this system, the application server consists of a set of PHP programs.

- Client tier: this is thin client which is a web browser running on the student computer. The client works as the user interface of the system. It mainly handles presentation of problems or other information that comes from the server. The information or problems presented in the web browser in this system are HTML pages.

5.3.2 Database Design

5.3.2.1 Design of Problem Base Database

ER Diagram and Tables

Based on the requirements for problem base and the above design, the problem base database is designed by using Entity-Relationship Modelling. Figure 5.3 is the logical view of problem base database.

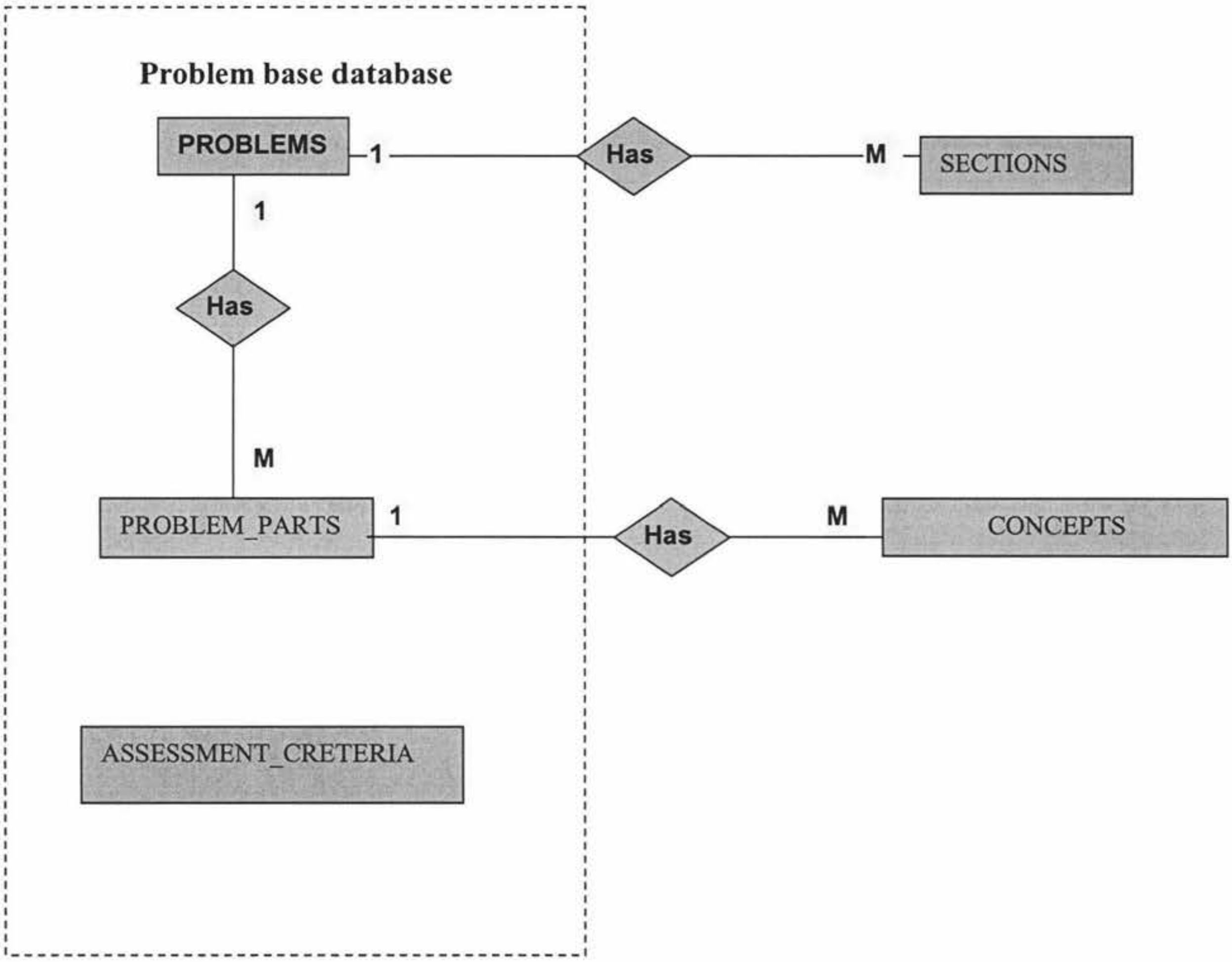

Figure 5.3: The logic view of problem base database (ER diagram)

The schema and contents of individual tables within problem base database are presented in Table 5.1.

Entity	Attribute	Date type	Constraint	Null value
Problems	ID	INTEGER	Primary key	No
	Name	CHAR(20)		No
	Subject	CHAR(50)		No
	Intended_User	CHAR(20)		No
	Add_Date	DATE		No
	Update_Date	DATE		No
	Author_Name	CHAR(20)		No
	Related_Sections_ID	CHAR(50)	Foreign key	No
Problem_Parts	ID	INTEGER	Primary key	No
	Name	CHAR(20)		No
	Difficulty_Degree	INTEGER		No
	Solution	CHAR(100)		No
	Problem_ID	INTEGER	Foreign key	No
	Related_Concept_ID	INTEGER	Foreign key	No
	Related_Section_ID	INTEGER	Foreign key	No
Assessment_ Criteria	ID	INTEGER	Primary key	No
	Assessment_Point	INTEGER		No
	Competence_Level	CHAR(30)		No
	Add_Date	DATE		No
	Modified_Date	DATE		No
	Author_Name	CHAR(20)		

Table 5.1: Entities and their attributes of the problem base database

Referential Constraints

In problem base database, there are some foreign keys that are primary keys from knowledge base database. These foreign keys link the problem base entity instances to the entity instances in the knowledge database. The 'cascade update' operation is performed in case of update of primary entity instances, i.e., if a primary entity instance is updated, and the primary entity contains the key value referenced by the corresponding foreign keys in the related entity instances, all related entity instances will be automatically updated. The 'cascade delete' is not enable, because one concept may be deleted, but the problem should not be changed or before changing, it needs to be confirmed.

5.3.2.2 Design of Knowledge Base Database

The knowledge base is needed to provide information for students. It mainly consists of problems-related sections, examples, and concepts. It may include wider knowledge than typical knowledge base in intelligent educational systems, because some advance students may want to exploit deeper knowledge than the course curriculum. It is ideal to have a searching knowledge base within this system (this will be one of the future works). The knowledge representation database also forms the foundation of problem base database.

The organisation of knowledge base in this system is different from typical knowledge base. The knowledge materials are organised in a content tree, and the units are related to problem base database. And the student model mainly will record the student's performances of solving problems.

Content Tree of Knowledge Base

The knowledge base materials are hierarchically organised into units of different levels in a content tree. From traditional textbook view, the structure of the knowledge base in this system is slightly similar to a content table of a book (Figure 5.4)

Chapter 1	
Section 1.1	learning section
Concept 1.1.1	
Concept 1.1.2	
Section 1.2	learning section
Concept 1.2.1	
Section 1.3	example section
Chapter 2	
Section 2.1	learning section
Concept 2.1.1	
Section 2.2	
Section 2.2.1	learning section
Concept 2.2.1.1	
Section 2.2.2	learning section
Section 2.2.3	example section
Chapter 3	
.	
.	
.	

Figure 5.4: Textbook content table view of knowledge base

There are two types of sections: content and non-content.

- Content sections: they are in the lowest level, such as section 1.1, section 1.2, and section 2.2.1 in Figure 5.4. These sections contain information related to problems and referred by problems, or example (e.g. section 1.3) showing how to use knowledge based on the information to solve problems. Under some of these sections, they may include some concepts that relate to parts of problems. Thus, the knowledge base also forms the foundation of the problem base database.

- Non-content sections: in contrast to content sections, they are not in the lowest level. These sections only act as the references for the material structure.

Each unit, such as section and concept, is stored as a HTML file that is presented to the student whenever the system decides the student need helps during problem solving processes.

Based on the above figure, the structure of knowledge base can be converted to a content tree (Figure 5.5).

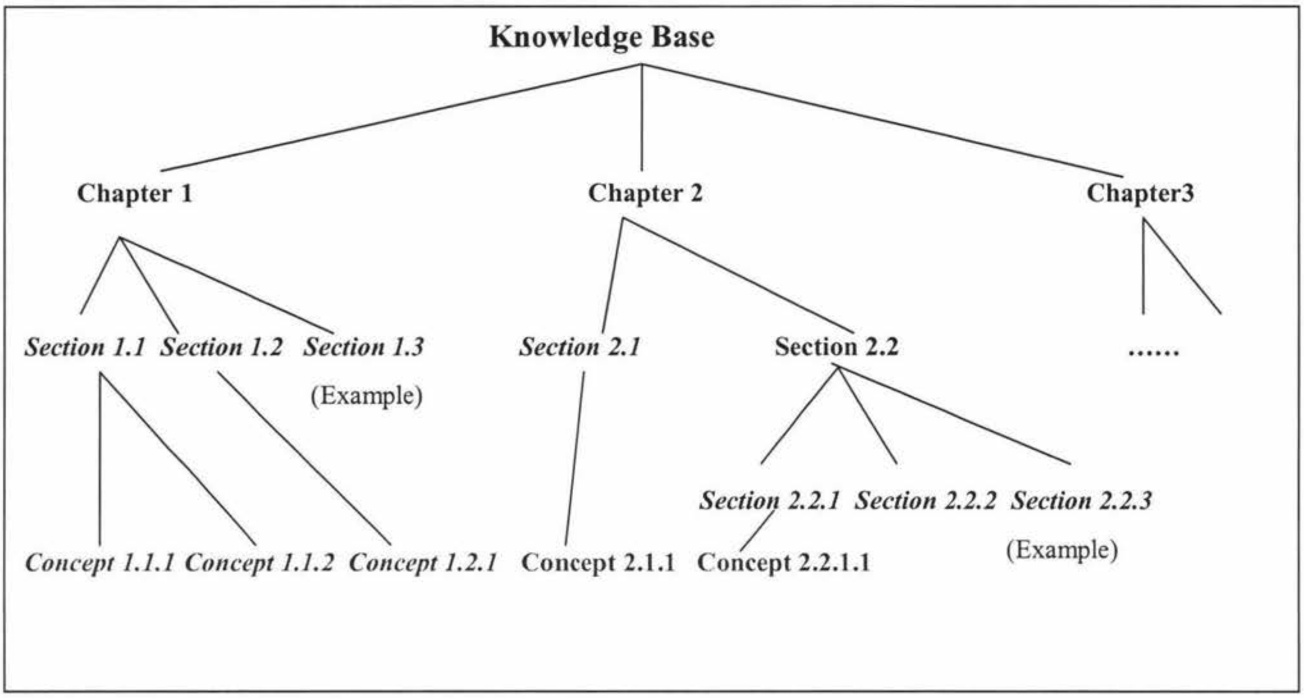

Figure 5.5: The content tree of knowledge base

Logic View and Tables of Knowledge Base Database

Figure 5.6 shows the entities and their relationship in knowledge base.

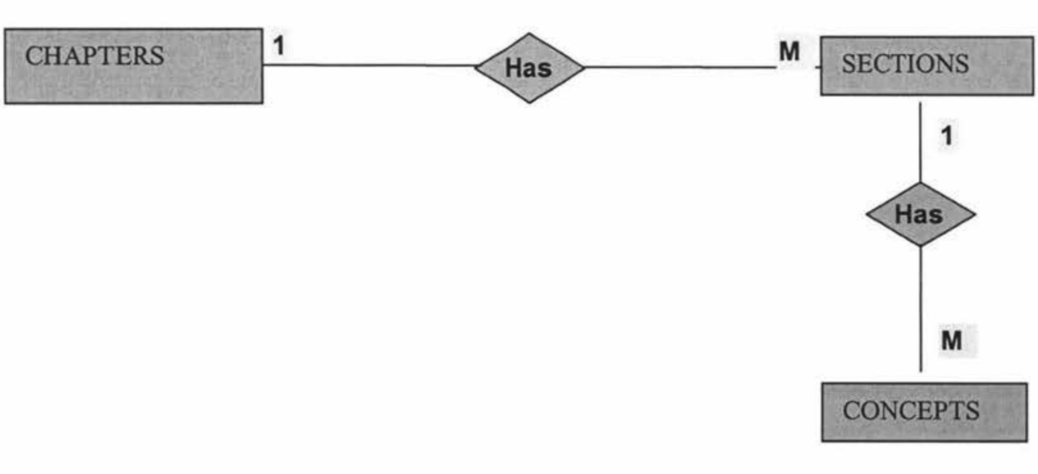

Figure 5.6: ER diagram of knowledge base database

The schema and contents of individual table within knowledge base database are presented in Table 5.2.

Entity	Attribute	Data Type	Constraint	Null Value
Chapters	Chapter_ID	INTEGER	Primary key	No
	Subject_Name	CHAR(30)		No
	Author_Name	CHAR(20)		No
	Add_Date	DATE		No
	Update_Date	DATE		No
Sections	Section_ID	INTEGER	Primary key	No
	Section_Name	CHAR(20)		No
	Parent_Section_ID	INTEGER		No
	Type	CHAR(20)		No
	Add_Date	DATE		No
	Update_Date	DATE		No
	Chapter_ID	INTEGER	Foreign key	No
Concepts	Concept_ID	INTEGER	Primary key	No
	Concept_Name	CHAR(20)		No
	Add_Date	DATE		No
	Update_Date	DATE		No
	Section_ID	INTEGER	Foreign key	No

Table 5.2: The schema and contents of individual table within knowledge base database

5.4 Summary

This system is implemented in the three tiers architecture, which especially introduces the problem base and student adaptivity mechanism into this PBS learning prototype. The system contains two databases: problem base, knowledge base. The knowledge base database is represented in a content tree, and also the foundation of problem base database that needs to refer to knowledge base based on the student performance during problem solving. Next chapter will describe the details of the system implementation.

Chapter 6

System Implementation

6.1 Introduction

This prototype is developed for Process Costing topic in Accounting domain. In this chapter, first some relevant technologies are discussed that were used to implement this system, such as PHP, HTML, and CGI. Then, the system workflow and main processes are described, followed by some example PHP code.

6.2 Relevant Technologies

The main technologies used to implement this system are PHP, CGI technique, HTML, Apache, MySQL (Figure 6.1). This section reviews these technologies.

6.2.1 PHP

6.2.1.1 The Origins of PHP

PHP was written by Rasmus Lerdorf in late 1994 (Bakken, 2000). It began as a simple way to track visitors to Rasmus Lerdorf's on-line resume, and could embed SQL queries in Web pages. Rasmus informally unleashed the first version – Personal Home Page Tools version 1.0. It only filtered input, replacing simple commands for HTML.

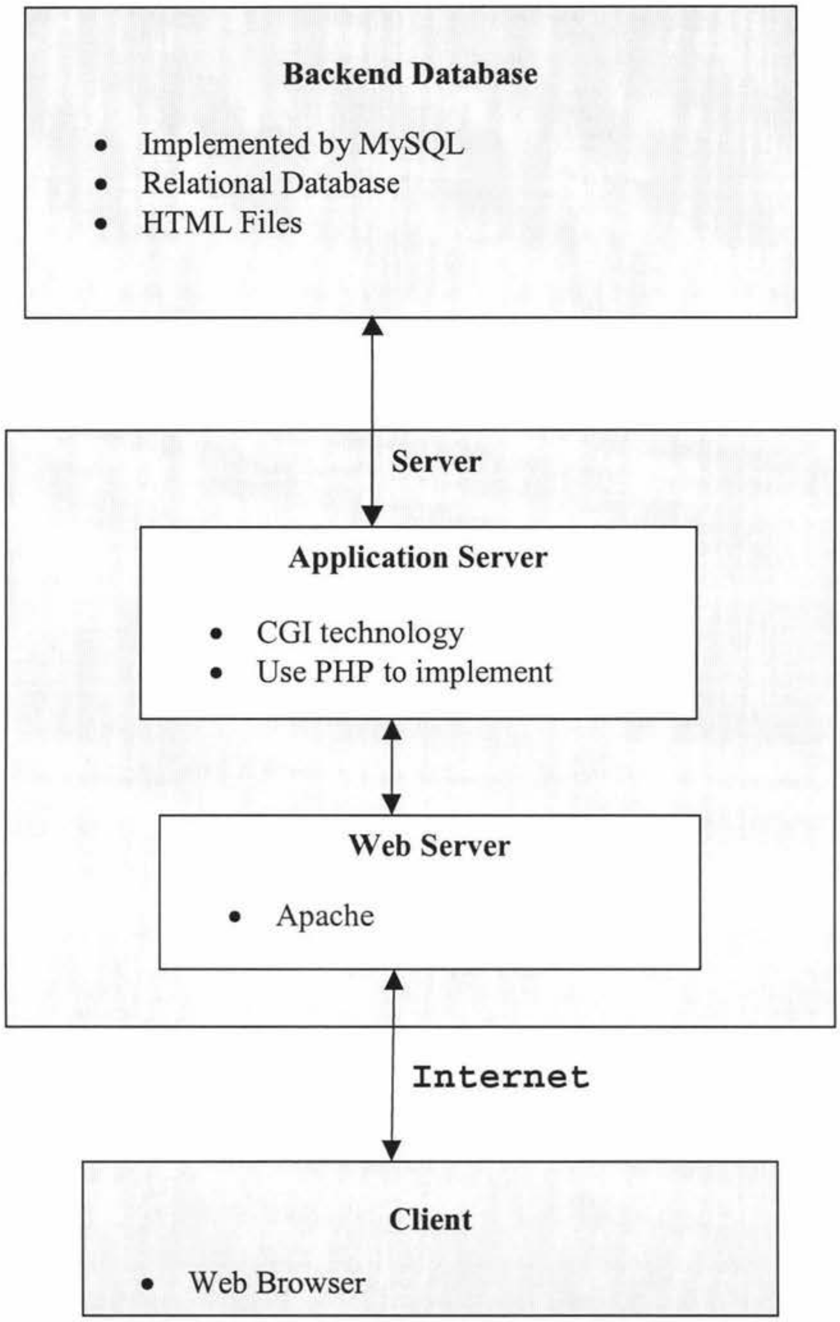

Figure 6.1: Related technologies used by the system

As its popularity grew, people wondered if it couldn't do more. Rasmus referred to other language parsers and created PHP 2.0. It allowed developers to embed structured code inside HTML tags. PHP scripts could parse data submitted by HTML forms, communicate with databases, and make complex calculations on the fly. It was very fast because the freely available source code compliled into the Apache Web serer. A PHP script executed as part of the web server process and required no forking, a frequent criticism of Common Gateway Interface (CGI) scripts. In 1996, PHP was a legitimate development solution and began to be used for commercial web sites. Clear Ink created the SuperCuts site (SuperCuts, 2002) and used PHP to create a custom experience for the Web surfer.

The most important step for PHP development is the release of PHP version 3.0 in June 1998. It was a state-of-the-art that combined the efforts of Rasmus, Zeav and Andi. PHP 3.0 version defined the syntax and semantics used in both versions 3 and 4. Currently there are over 1.4 million web servers running PHP scripts (Meloni, 2001)

6.2.1.2 Features of PHP

PHP is a server-side, cross-platform, HTML-embedded scripting language (Davis, 1999). Much of PHP's syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

The main features of PHP are:

- Faster to code and execute: PHP eliminates the need for numerous small cgi programs by allowing developers to place simple scripts directly in HTML files. It also makes it easier to manage large web sites by placing all components of a web page in a single html file. For example, one application with PHP can be used to generate all the pages required to display items from a database. This eliminates redundant page generation, simplifying maintenance. PHP can be set up as one module of the Web server, so it is useful for dynamic Web page design, quicker response and transparency to the end user.

- The same PHP code runs unaltered on different web servers and different operation systems: PHP runs on UNIX, Windows, and Macintosh. PHP is designed to integrate with the Apache web server, which is the most popular web server on the Internet. But PHP works equally well with other web servers, including Microsoft's Internet Information Server. Scripts may be moved between server platforms without alteration.

- Built-in features: PHP has a large set of built-in support for numerous databases (including Access, LDAP, Oracle, and MSSQL), networking support, zip archiving and an excellent set of built-in functions.

- Freely available: It can be freely downloaded online from PHP group web site (PHP Group, 2002). PHP is an excellent alternative to such similar programming solutions as Microsoft's proprietary scripting engine ASP and Allaire's rather expensive ColdFusion.

6.2.1.3 Mechanisms of PHP

Figure 6.2 shows the PHP 4.0's architecture. The Zand engine – PHP parser is a self-contained component, and PHP function modules are also self-contained. This structure makes the improvement and maintenance easily.

The SAPI is a web server abstraction layer that greatly simplifies the task of adding native support for new web serves. SAPI currently has server implementations for Apache, Roxen, Java (servlet), ISAPI (Microsoft IIS and soon Zeus), AOL server and CGI. All of PHP's functions are part of one of the layers with a side facing up in the architecture figure. Most functions such as the MySQL support, are provided by an extension, which is optional and can be linked into PHP at compile time or built as dynamically loadable extensions that can be loaded on demand.

The scenario of PHP working with Web server is: if the request from a client is for an HTML file, the web server will simply find the file, and tell the browser to expect some HTML text, and then send the contents of the file; if a web server gets a request for a CGI, the PHP script will be passed through the PHP engine and executed which will give the web server HTML text.

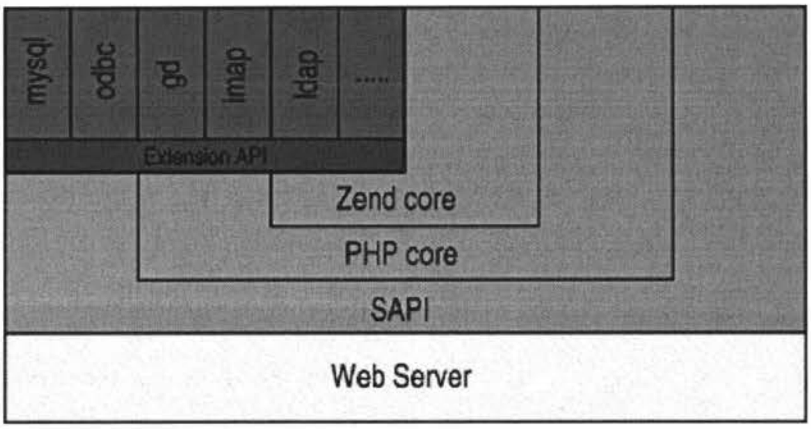

Figure 6.2: The architecture of PHP 4.0 (Bakken, 2000).

6.2.2 HTML - Hypertext Markup Language

Web pages are written as ASCII text files in a language called the Hypertext Markup Language, HTML. HTML consists of a set of instructions call tags that tell web browsers how to display the text in the page. Its file name ends in .html or .htm to indicate that it is an HTML file. The HTML tags that describe the way the document looks are enclosed in angle brackets<>. The tags are easy to read. If you want to create a title, for example, the tag instruction is enclosed in brackets, the actual text for the title is sandwiched between the marker that starts the instruction, <TITLE>, and the tag that ends the instruction, </TITLE>. The following line is called a TITLE element.

<TITLE> On-line bibliography</TITLE>

When the browser sees this instruction, the title will be printed in the bar at the top of the browser's window as a title for the page.

Links

Links or hyperlinks are images and text enclosed within anchor tags to produce point-to-point references to other document. The links can be created to connect one section of a web page to another resource on the Internet. The text is usually underlined and colored blue. Examples are as follows:

<A HERF="http://www.apache.org/">Apache</A>

HREF is assigned the URL, the address of the file that will be linked if the user clicks on the word *Apache*

Forms

Forms are HTML documents that allow the client to send data to Web server for processing. Normally the three types of forms: INPUT, SELECT, and TEXT.

The following example is an input form:

```
<form action="register.php" method="post">
<input type=submit value=Register name="submit">
</form>
```

The *action* parameter specifies the CGI program that will be running when the form is submitted, and the *method* defines either GET or POST method used. The *input type* can be "text", "password" or "submit" and the *value* is the string showed on the submit button and the *name* is the button's name.

Samples of HTML Tags and what they do

Table6.1 describes some samples of HTML tags and their function.

Tag Element	**Function**
<A HREF=http://www.apache.org>	Where this document is stored.
<HTML>document</HTML>	Found at the beginning and end of the document, indicating an HTML document.
<HEAD>heading</HEAD>	First element inside the document. Normally contains the title and is displayed outside of the main window of the browser.
<TITLE>title</TITLE>	Title of the element; displayed outside the document text in a window frame.
<BODY>contents</BODY>	Contains all the text and other objects to be displayed.
<H1>heading type</H1>	Heading elements, for levels 1 through 6, create bold face headings. The largest, outermost heading is H1.
<P>text</P>	Paragraph tag. Marks the beginning of a paragraph. Inserts break after a block of text.
<B>text</B>	Bold
<I>text</I>	Italic
<U>text</U>	Underline
<A HREF SRC="URL">	Create hotlink to a resource at address in URL on the web.
<IMG SRC="URL">	Load an image into a web page. URL is the image file.

Table 6.1: Samples of HTML tags

6.2.3 CGI – Common Gateway Interface

CGI stands for Common Gateway Interface. It is a standard for interface external applications with information servers, such as HTTP (Hypertext Transfer Protocol) or Web servers. Its function is allow the WWW server to go beyond its normal boundaries for retrieving and accessing information from external databases and files. It is a specification that defines how data can be transferred from the script to the server and from the server to the program.

Gateway programs, called CGI scripts can be written in any programming language, but PHP has become very popular now because its flexibility and easy to use. In this project, PHP is used to write CGI scripts.

In a HTTP operation without CGI programs, the client computer sends an HTML request of a web page to the WWW server, the server finds the page and sends the HTML back to the client. The client browser interprets the HTML code and displays it.

In CGI, HTML coded elements on the page cause data to be sent to the WWW server along with the HTML request. This request is passed onto a program, which may retrieves data from server side database, analyses and acts upon it, and then passes an HTML text to the web server that communicates with the client. CGI programs are executed in real-time, so it produces the dynamic web page.

6.2.4 MySQL

The MySQL server is one of the fastest SQL (Structured Query Language) database servers currently on the market and the most popular Open Source SQL database. MySQL is developed by T.c.X DataKonsultAB and is available for download from MySQL AB Company web site (MySQL AB Company, 2002).

MySQL is a relational database management system. A database is a structured collection of data. A relational database is one type of databases that stores data in separate tables and links tables by defined relations making it possible to combine data from several tables on request. To add, access, and process data stored in a computer database, a database management system, such as MySQL, Oracle, DB2 is necessary, which plays a central role in computing, as stand-alone utilities, or as parts of other applications.

MySQL is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and several programming interfaces.

PHP and MySQL are the world's best combination for creating data-driven sites (Merrall, 1999)

6.2.5 Apache

The Apache is a http server that is developed by the Apache Group in 1995 (Apache Group, 2002). Its main features are:

- is a powerful, flexible, HTTP/1.1 compliant web server
- implements the latest protocols, including HTTP/1.1
- is highly configurable and extensible with third-party modules
- can be customized by writing 'modules' using the Apache module API
- provides full source code and comes with an unrestrictive license
- runs on Windows NT/9x, Netware 5.x, OS/2, and most versions of Unix, as well as several other operating systems.

The main technologies used in this project have been discussed above. These technologies form two popular and ideal web based applications development environments: LAMP (Linux, Apache, MySQL and PHP) for Linux operation system, and WAMP (Windows, Apache, MySQL and PHP) for windows operating system (Atkinson, 2001).

6.3 System Implementation

6.3.1 System Working Processes

Figure 6.2 outlines the process flow of this system. The process detail of inferring student's results and its algorithm are discussed as below.

From figure 6.2, after student logins successfully, the system presents the main menu page (figure 6.3), then presents the problem developed from the real world on how to solve an interactive process costing problem when student clicks the *Process Costing* link. The example of problem is shown in figure 6.4.

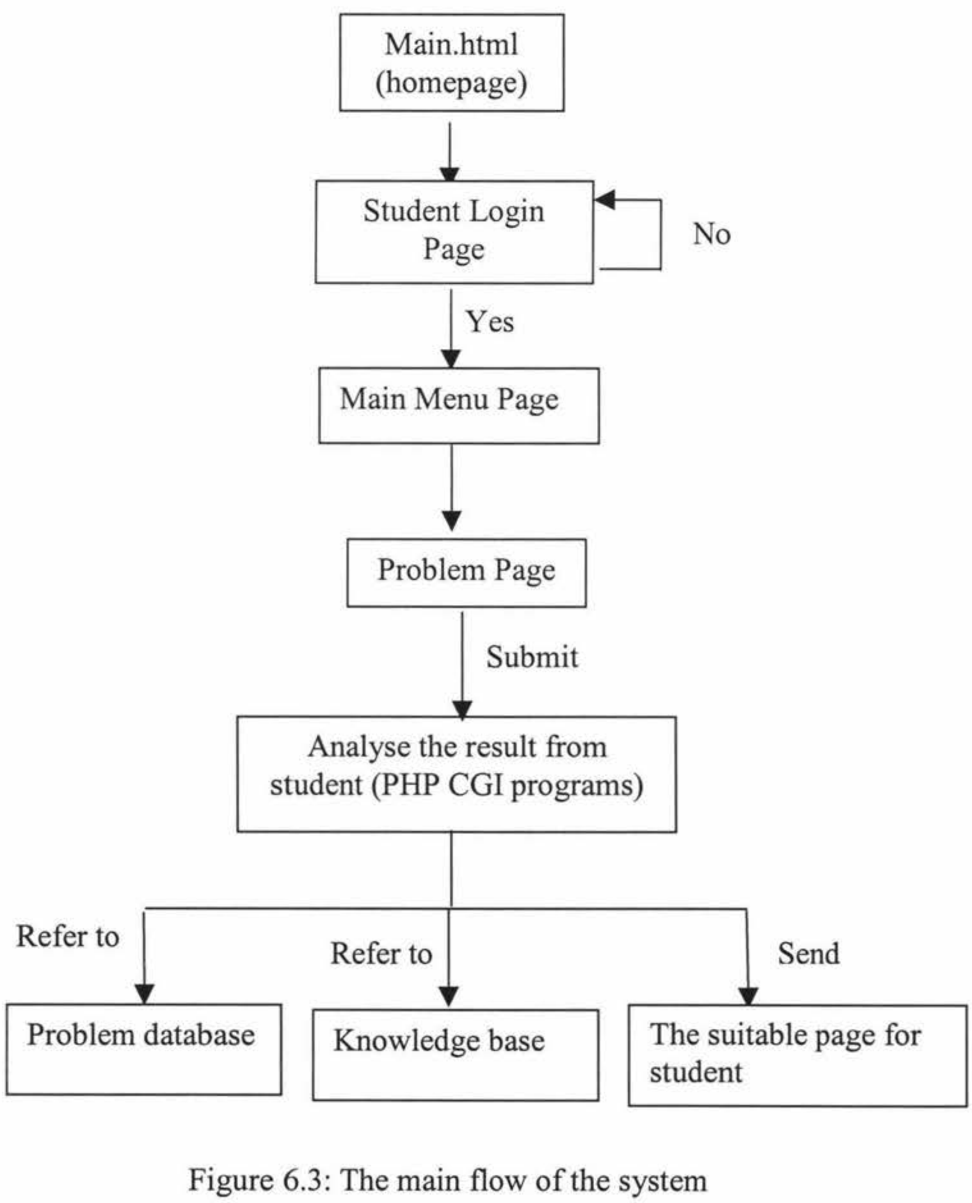

Figure 6.3: The main flow of the system

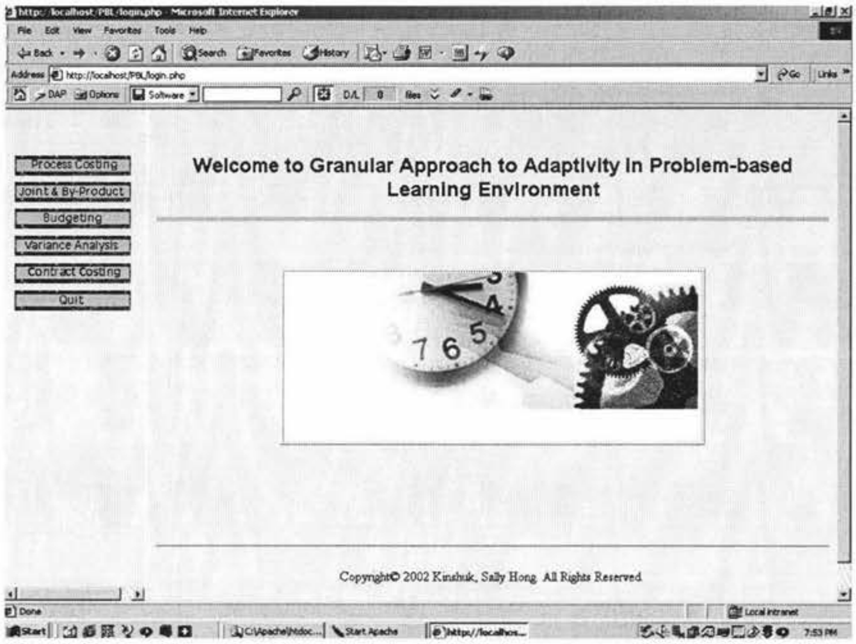

Figure 6.4: Main menu page

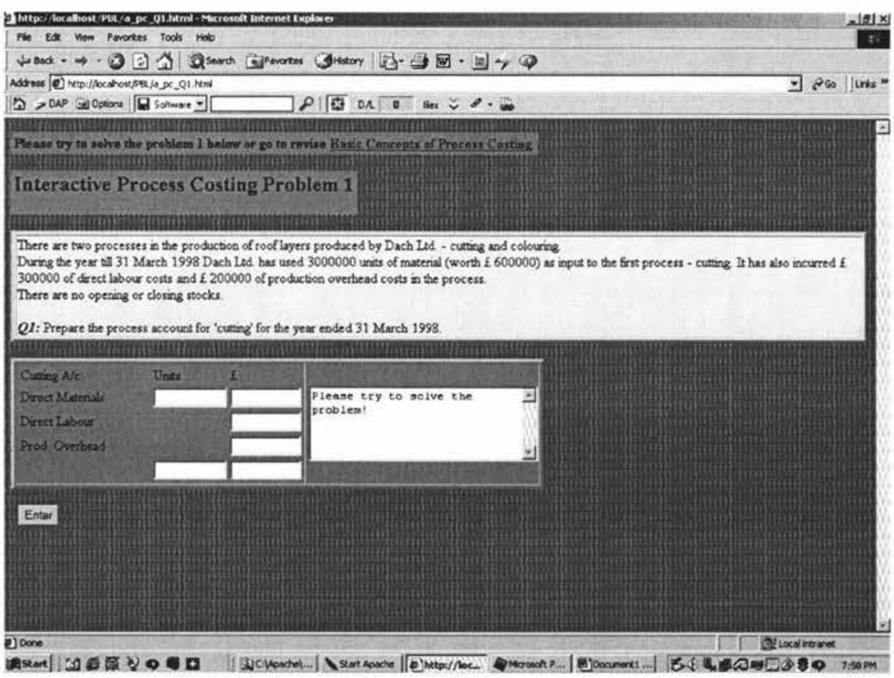

Figure 6.5: Example of one problem for student

Students work on the problem and answer each part of the problem. When they feel satisfied with the answers, they submit all answers to the system. The server receives the answers for the problem from the client. PHP program then processes the answers against the database, whose algorithm is described below based on the above example.

6.3.2 Example PHP Code

This system was implemented using PHP. Some examples of PHP program within this system are presented as follows.

Backend.php: It is a class that contains most main function definitions, which are called by others modules. The main function within the backend.php class is described below.

- *Question1_check ($DM_units, $DM_pounds, $DL_pounds, $PO_pounds, $Total_units, $Total_pounds):* It is a function that takes variables from question1.html in which a student fills in answers, and compare the student's answers with solutions stored in the database, then calculate the total points the student get. This function returns the total points for the student in solving the problem. The sample code is below:

```
Question1_check($DM_units,   $DM_pounds,   $DL_pounds,   $PO_pounds,
$Total_units, $Total_pounds) {
   mysql_connect($this->server, $this->db_user, $this->db_pass);
   mysql_select_db($this->database);
   $query  =  mysql_query("select  *  from  problem_parts    where
          problem_id = '$ problem_num");
   mysql_close();
   list  ($solution1,  $  solution2,  $  solution3,  $  solution4,
   $solution5, $solution6) = mysql_fetch_row ($query);

   if ($DM_units == $solution1) { $this->point1=10; }
   if ($DM_pounds == $solution2) {$this->point2=10; }
   if ($DL_pounds == $solution3) {$this->point3=10; }
   if ($PO_pounds == $solution4) {$this->point4=20; }
   if ($Total_units == $solution5) {$this->point5=20; }
   if ($Total_pounds == $solution6) {$this->point6=30; }

   $this->TotalP=$this->point1+$this->point2+$this->point3
         +$this->point4+$this->point5+$this->point6;

   return $this->TotalP ;
}
```

- *Decision ($totalpoint)*: The function takes the total points variable and compare with the criteria defined by teachers and stored in the database, then return the student's level. The code for this function is presented as follow:

```
function decision($totalpoint){
   mysql_connect($this->server, $this->db_user, $this->db_pass);
   mysql_select_db($this->database);

   $query = mysql_query("select * from assess_criteria where
         problem_id = '$ problem_num");
   mysql_close();
   list ($begin, $medium1, $medium2, $advance) = mysql_fetch_row
   ($query);

   if ($totalpoint< $this->begin) {
         $level =1;
         return $level;
   }
   if ($totalpoint>= $this->begin && $totalpoint< $this->medium1){
         $level =2;
         return $level;
   }
   if($totalpoint>=$this->medium1 && $totalpoint< $this->medium2){
         $level =3;
         return $level;
   }

  if($totalpoint>=$this->medium2 && $totalpoint<= $this->advance){
         $level =4;
         return $level;
   }
}
```

- *Register (username, password, password2, name, email, age, sex, school):* It is a function that takes variables from register.html, in which a student fills in his/her profile. It then checks if each variable matches the field criteria, if there are some fields filled by invalid data, the function returns an error and reminds the student to correct them. Otherwise, the function compares the student's user name and email with student data stored in the database, if it is a new registration, then add the data into the table in the database.

```
function  register  ($username,  $password,  $password2,  $name,
$email, $age, $sex, $school) {
   if(!$username ||!$password ||!$password2 ||!$name ||!$email ||
   !$age ||!$sex || !$school) {
         return $this->error[14]; }
   else {
         if (!eregi("^[a-z ]+$", $name)) {
               return $this->error[8]; }
         if(!eregi("^([a-z0-9]+)([._-]([a-z0-9]+))*[@]([a-z0-
           9]+)([._-]([a-z0-9]+))*[.]([a-z0-9]){2}([a-z0-9])?$",
           $email)) {
               return $this->error[4]; }
         if (ereg("[^0-9]", $age)) {
               return $this->error[10]; }
         if ($sex != "Male" && $sex != "Female") {
               return $this->error[11]; }
         if (!eregi("^[a-z0-9 ]+$", $school)) {
               return $this->error[9]; }
         if (strlen($username) < 3) {
               return $this->error[1]; }
         if (strlen($username) > 20) {
               return $this->error[2]; }
         if (!ereg("^[[:alnum:]_-]+$", $username)) {
               return $this->error[3]; }

         if ($password != $password2) {
               return $this->error[0]; }
         if (strlen($password) < 3) {
               return $this->error[5]; }
         if (strlen($password) > 20) {
               return $this->error[6]; }
         if (!ereg("^[[:alnum:]_-]+$", $password)) {
               return $this->error[7]; }

         mysql_connect($this->server,$this->db_user,
               $this->db_pass);
         mysql_select_db($this->database);
```

```
        $query = mysql_query("select id from login where username
                = '$username'");
        $result = mysql_num_rows($query);
        if ($result > 0) {
              mysql_close();
              return $this->error[12];
        }

        $query = mysql_query("select id from authlib_data where
                  email = '$email'");
        $result = mysql_num_rows($query);
        if ($result > 0) {
              mysql_close();
              return $this->error[13];
        }

        $hash = md5($this->secret.$username);
        $is_success = mysql_query("insert into authlib_confirm
                      values ('$hash', '$username', '$password',
                      '$name', '$email', '$age', '$sex',
                      '$school', now())");
        mysql_close();
        return 2;
}
```

Question1.php: when a student finishes the problem 1 and summits it, this program will be fired. It calls the relevant functions in the class *backend.php* and refers to the various html pages accordingly.

Below is a sample code from question1.php.

```
<? require("backend.php");
      $totalpoint = $authlib->question1_check ($DM_units, $DM_pounds,
                    $DL_pounds, $PO_pounds, $Total_units,
                    $Total_pounds);

      $level = $authlib->decision($totalpoint);

      if ($level == 1){ include("a_pc_Q1_Begin.html"); }
```

```
    if ($level == 2){ include("a_pc_Q1_Medium1.html"); }
    if ($level == 3){ include("a_pc_Q1_Medium2.html"); }
    if ($level == 4){ include("a_pc_Q1_Advanced.html"); }
    if ($level == 5){ include("a_pc_Q2.html"); }
?>
```

- First line is to import the backend.php class.
- Second line is to call a function *question1_check* in Object *authlib*, and variables are passed in from question1.html. The function *question1_check* returns the result that is assigned to variable *totalpoint.*
- Third line is to call a function *decision* in Object *authlib*, and pass the variable *totalpoint* into the *decision* function that returns the result to variable *level.*
- Finally, a corresponding html page will be opened according to the level.

6.4 Summary

The related technologies used to implement this prototype were discussed in this chapter. PHP is getting popular in web-based application, and WAMP (Windows, Apache, MySQL and PHP) for windows operating system used in this project is one of the two popular and ideal web based applications development environments, which is this one and LAMP (Linux, Apache, MySQL and PHP) for Linux operation system. Then the main processes within the prototype are discussed, and some examples of PHP code in the system implementation are presented. In the next chapter, the student adaptivity in this system is discussed.

Chapter 7

Student Adaptivity in This System

7.1 Introduction

As presented in Chapter 6, the prototype of web-based problem-based learning (PBL) for accounting with student adaptivity function has been designed and implemented. The student adaptivity in this PBL system is able to address the limitation of the typical PBL learning environment, where students easily loose focus and get frustrated by the lack of adequate guidance. This chapter presents how the student adaptivity works in this system.

7.2 Monitoring Student's Performance

When a student uses this system, she/he is presented with a problem. The student works on the problem, answers each part of the problem, and then submits the answers to the system. The inference engine within the system refers to the problem base in the backend database. The problem base consists of a set of different levels of real world problems, the solutions for each part of the problems, assessment criteria that includes weights for various parts in a problem that are of varying difficulty, and definition of student's level. The content of problem base is provided and defined by the teachers. The inferring steps includes:

- Matching the student's answers with standard answers provided by the teachers;
- Calculating the final result for the student and deciding the student's level based on the level definition created by the teachers.

In fact, the quality of problem base is the core of this system, which largely depends on teachers' experience.

7.3 Presenting Proper Guidance

Based on student performance, the system adapts itself to provide proper help to students. For a beginner, the system provides more direct and concrete helps to keep the student on the track and to encourage him/her to carry on. But because the PBL environments aim to foster learner's problem-solving and self-directed learning skills, the system gradually fades its tutor role with student's increasing progress and improving level, for example, only pointing an abstract direction. Ideally, the system offers enough help to encourage students to continue, but leaves enough difficulty and challenge space to students. For example, in this system:

- If the student got less than 50 points in a problem out of 100, then the system infers the student at the beginning level of the assessment criteria defined by teachers. The system in this case provides very concrete guidance to the student, such as presenting the analysis of his/her answers as feedback to the student, the related information from introduction and basic concepts to examples, and then recommends the student to review the relevant content and try the problem again (figure 7.1). From this feedback and recommendation, the student is able to decide his/her next step and carries on by opening the links provided by the system. With this system, the beginner will not loose focus or direction or get frustrated without enough help. However, the system only narrows the information scope and still leaves options for the student.

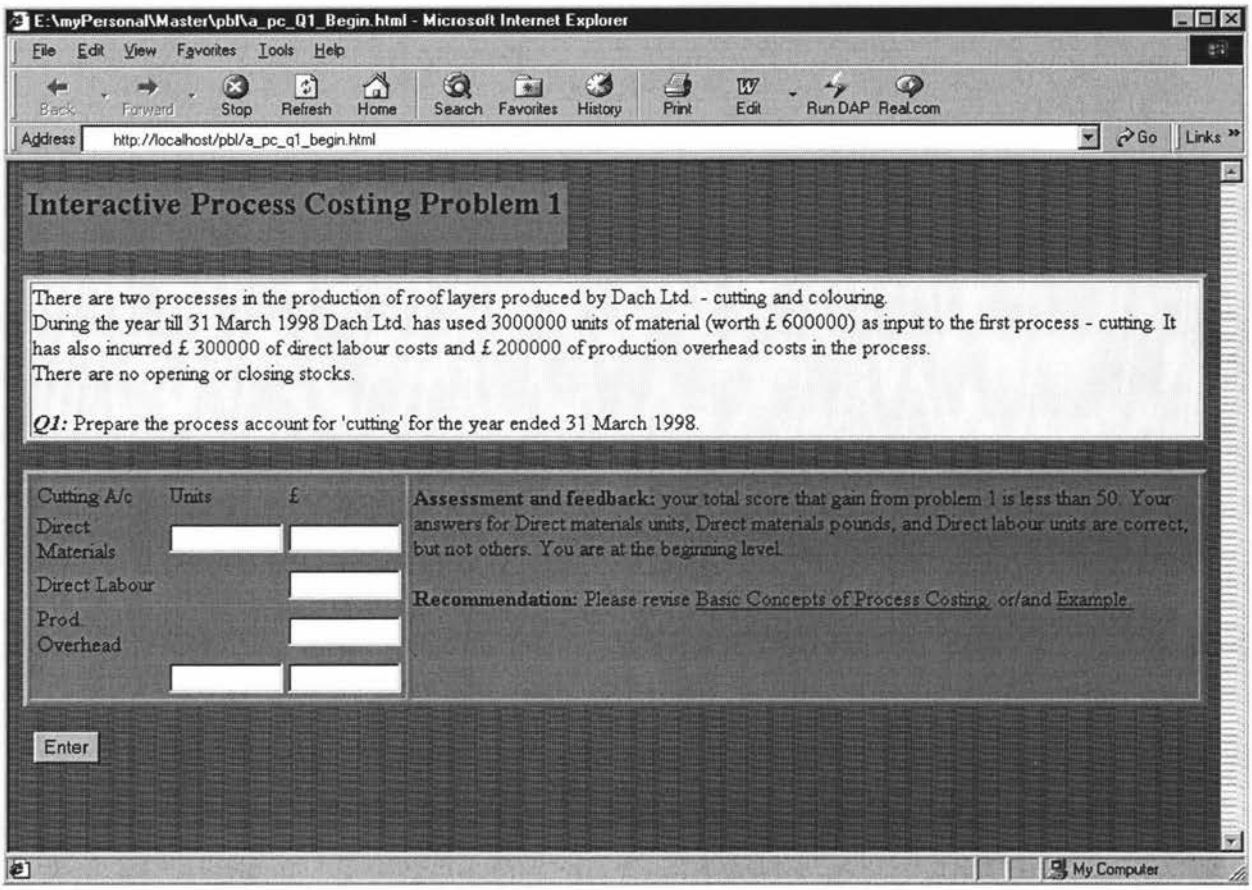

Figure 7.1: The screenshot for the beginning level student

- If the student got 51-70 points in a problem, then the system infers the student at the medium-one level of the assessment criteria that is defined by teachers. In this case, the system infers that the student has understood the concepts related to the problem, but has difficulty at using concepts to solve the problem. Therefore, the system presents the example of the problem, and recommends the student to review the example and try the problem once again.

- If the student got 71-90 points in a problem, then the system infers the student at the medium-two level of the assessment criteria that is defined by teachers. In this case, the system infers that the student has some knowledge to solve the problem, but not yet mastered the knowledge completely. Thus, the system gives feedback of the student's performance, but not as detailed as the feedback for the beginning level students. The system also gives recommendations to the student, such as reviewing example (figure 7.2), and the students can know where and why their answers are wrong. If the system still provides some basic information and very detailed analysis as feedback to the student, the student may get bored.

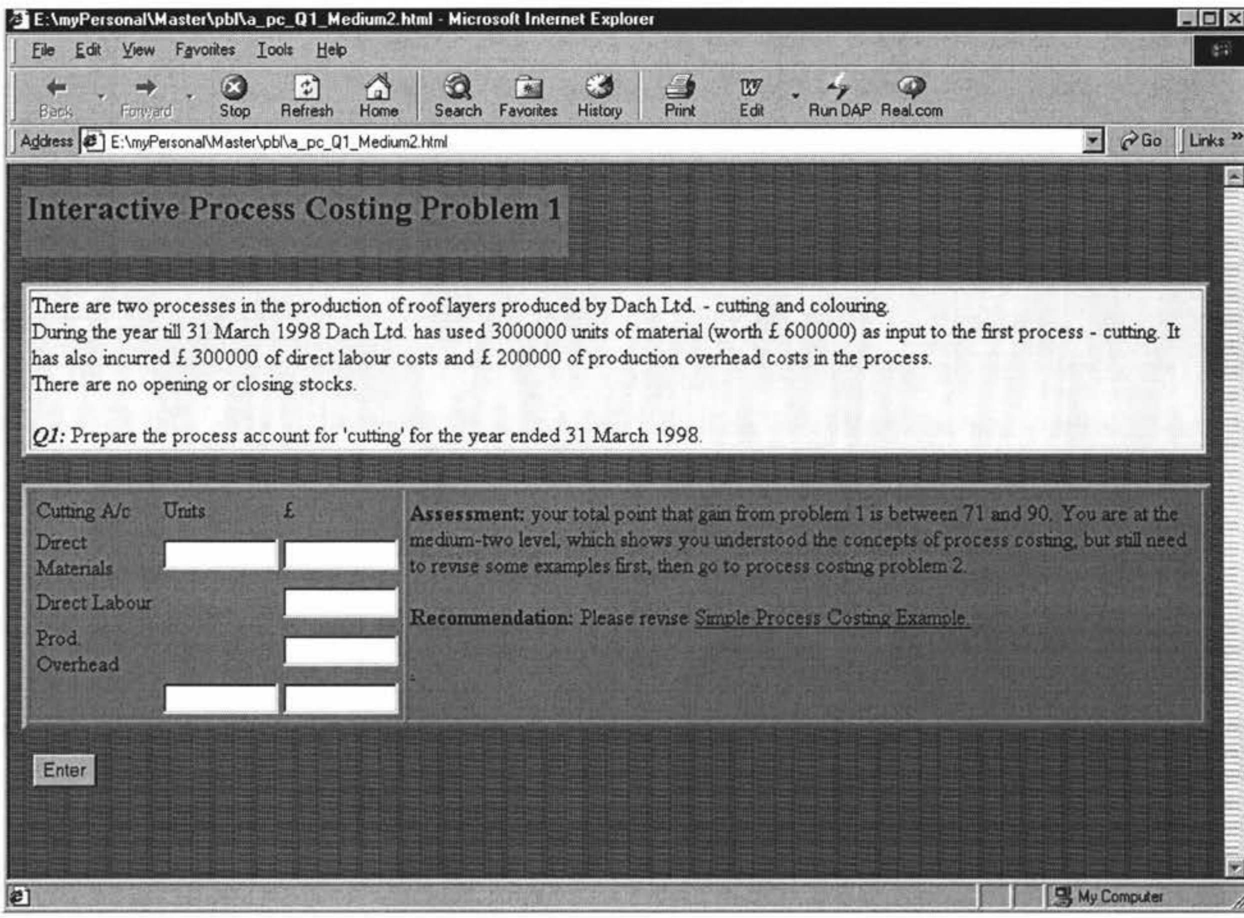

Figure 7.2: The screenshot for the medium-two level student

- If the student got more than 90 points in a problem, then the system infers the student at the advance level of the assessment criteria that is defined by teachers. In this case, the system infers that the student has adequate knowledge related to the problem. In that case, the system just sends the student to a more complicated problem (figure 7.3).

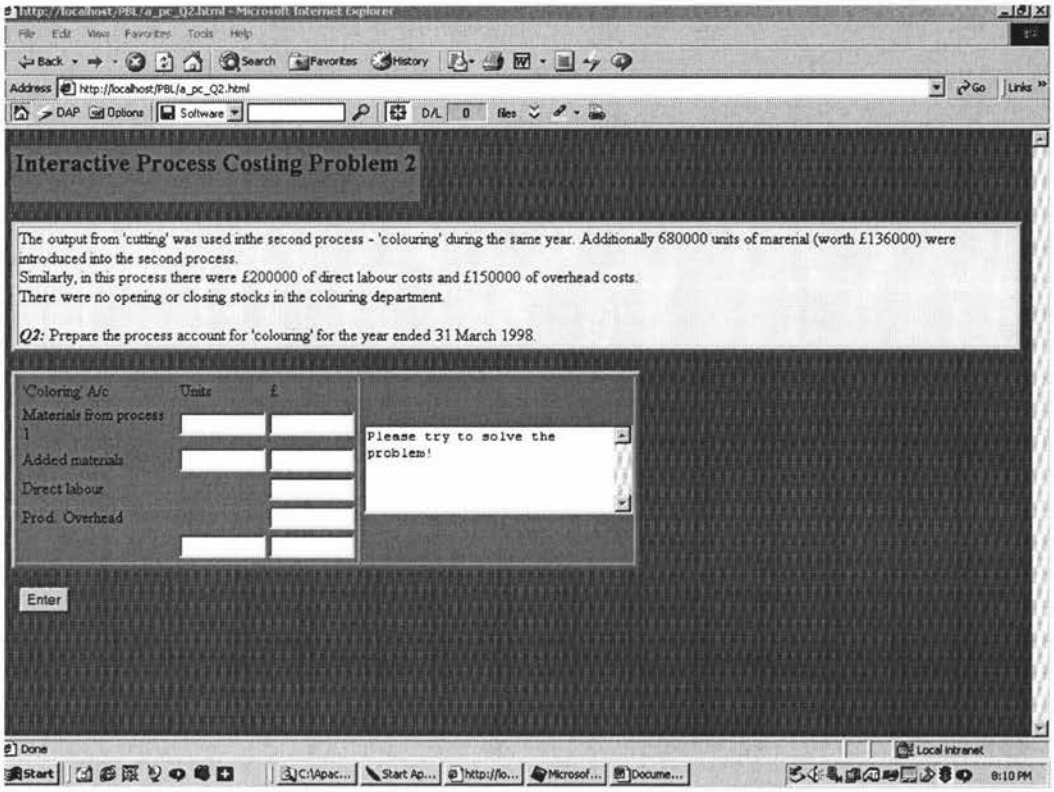

Figure 7.3: The continuing screen for the student who reaches the advance level in solving the problem 1

7.4 Algorithm in the System

7.4.1 Assumptions

Before we discuss the algorithm, we assume that the points for each part of the problem 1 defined by teachers as are as follows:

- Direct materials units – 10 points
- Direct materials pounds – 10 points
- Direct labour units – 10 points
- Prod. Overhead – 20 points
- Total units – 20 points
- Total pounds – 30 points

Assessment criteria defined by the teachers for this problem is as follows:

ID	ASSESSMENT_POINT	COMPETENCE_LEVEL
1	0-50	beginning
2	51-70	Medium_one
3	71-90	Medium_two
4	91-100	Advance

7.4.2 Algorithm

The basic algorithm in the system is presented as follows.

```
Check each answer for each corresponding part of the problem
against the solution stored in the problem base

Calculate the total points student earned

If total points < 50 then
      It is beginning level
      Revise: concepts, example (figure 7.1)
      Do the exercise again
      Then assessment again
If total points = 51-70 then
      It is medium_one level
      Revise: example
      Do the exercise again
      Then assessment again

If total points = 71 - 90 then
      It is medium_two
      Revise the example (figure 7.2)
      Then go to next question

If total points = 91 - 100 then
      It is advance level
      Go to the next question (figure 7.3)
```

7.5 Evaluation of the Prototype

The evaluation described here was performed to investigate the claim that the ideas introduced in the research improve the effectiveness and efficiency of students using PLB learning environments. Currently, this is an early evaluation and the result of the evaluation will be referred when enhancing the system. A formal evaluation of the impact of the approach suggested by the project will be designed and conducted in the near future.

To conduct the empirical evaluation, there are 6 people were invited to use the system and complete a questionnaire. This group of subjects consists of: three undergraduate and three postgraduate students aged between 25 - 35. Within undergraduate students, 2 are female and were from accountancy department; and 1 male and from finance, banking and property studies department. Three postgraduate students include 1 female who was from accountancy department and 2 male from information systems department.

The questionnaire for completing by the 6 subjects included 4 questions. The details of feedback are described as follows.

Q1: Are you pleasant to use the system? Why?

All subjects were satisfied by using the system and fairly like it. They recognized that it is very straightforward to know or learn how to use the system and the user interface are fairly conformable. There is no any Meta data overloading problems. In term of navigation of the content, they thought the adapted content presentation eases their navigating within the content and saves them lots of time

Q2: Is the system useful for students? Why?

Four students reckoned that the system is quite helpful for students, especially for the distance learning or extramural papers. The adaptive capability in the system can reduce the opportunities of losing direction from students. In particular, those three students who were not from accounting background pointed out that most time they easily found some helpful links when wondering, such as related sample or concepts. Therefore, the system can improve students' learning confidence and interest. One student from accountancy department felt the information (content and links) presented to her is too limited, and hoped she can have some more options when helps were needed.

Q3: In term of improvement of the system, please give a suggestion?

Three students reckoned the links provided by the system are quite limited, and suggested the system should provide some more optional links so that student can have wide learning space, in particular advance students. Two subjects proposed that it is important to provide effective communication channels between students and instructors; especially it is even crucial in cases of distant learning. One student thought it would be a good idea to have a simple demo to illustrate the basic idea of the system or the course.

Q4: What roles will the system play in their learning process?

3 subjects expected the system will provide a very good supplement when they work in a solving problem environment, it can reduce the difficulty confronted during learning processes. 1 student reckoned the system will be helpful when students study from distance or learn their own. 1 student preferred to study with learner groups or under instructing by tutors.

According to the feedbacks received from the empirical evaluation, the design and implementation will be reviewed and a number of enhancements for the system will be investigated in future work.

7.6 Summary

The system achieves the student adaptivity by adapting itself according to student performance. Based on the assessment criteria defined by the teachers and student's level at solving problems, the system provides the flexible student adaptivity to the students. On one hand, this system provides more concrete feedback and information to beginners to keep them on the track. On the other hand, the system offers less detailed feedback to higher level students to improve their self-directed learning and problem-solving skills. The next chapter concludes this thesis and describes future work.

Chapter 8

Conclusion

8.1 Introduction

This chapter summarises this research. Firstly, the main contributions of this thesis are described, and then the possible future work, for improving the existing prototype is discussed.

8.2 Contributions

In this research, the prototype of web-based problem-based learning (PBL) for accounting was investigated and designed to facilitate student adaptivity function into computer-based PBL learning environment. Thus, the system is able to enhance the effectiveness and efficiency of students using PBL learning environment.

The main goals achieved in this work are: the architecture of PBL with student adaptivity was designed, and the prototype based on the architecture was implemented. The implementation of the system was done in such a way that the problems, assessment criteria and knowledge base could be easily authored, so the database design, system design and implementation strategies facilitate the future extension of the system to enable authoring of the domain content.

The main contributions in this research are described below.

8.2.1 The Architecture and Mechanism

This project resulted in an architecture and mechanism of PBL with certain student adaptivity. This result not only can be applied in pure PBL leaning environment, but also can be implemented in other intelligent educational systems to enhance their assessment or testing module.

8.2.2 The Implementation of Prototype

Based on the above architecture, a web-based prototype was successfully built by using PHP, Apache, MySQL and HTML. The domain of the prototype is accounting, especially costing processes. The prototype attempts to help students to improve their problem solving skills by using real world problems. The system based on student's performance provides help or domain content to student. The student will not easily loose focus during learning in PBL environment.

8.2.3 Abstraction of Database Design from the Implementation

The implementation of the prototype allows addition of authoring facilities in future. The database is abstracted from the system implementation, that means the problem base, knowledge base, and assessment criteria can be modified or added and will not influence the system. In the future work, we will briefly discuss how to add authoring module into this system. This database is implemented in MySQL that is a powerful database server, so that the database is able to extend easily.

In summary, this research has accomplished its goal of introducing student adaptivity technologies into PBL learning environment, and developing a prototype to demonstrate the idea. The system offers a problem-based learning environment with certain student adaptivity for accounting students to learn costing process based on the real world problems.

8.3 Future Work

The prototype is built to exploit the student adaptivity technologies within problem-based learning environment. Since it is designed by keeping extensions in mind, the system can be extended further in the following aspects.

8.3.1 Adding Authoring Tools

An authoring component can be added into this prototype to facilitate the authoring of problems, assessment criteria, and knowledge base. Application as part of this prototype could be developed to provide interface to access the problem base, and knowledge base. Teachers can then easily add and modify problems, and their assessment criteria. In addition, teachers may also be able to author the knowledge base according to the needs of their teaching.

8.3.2 Adding Student Model

A student model could be added to complete the student adaptivity of the prototype. The student model could record the student's behaviour and preference to improve the adaptivity functionality in the system.

8.3.3 Adding Collaborative Learning Facilities

Problem-based learning benefits from collaborative group learning and increasing interaction between students and faculty (Koschmann, et al., 1990). To move to a complete PBL system, the system should provide tools or channels to enable the discussion among student and between student and teachers. This direction of research will also exploit the true potential of the web.

References

ADE. (2002). *Advanced Distance Education* http://www.isi.edu/isd/ADE/ade.html (last accessed 6/5/02).

Albanese, M. and Mitchell, S. (1993). *Problem-based learning: A review of the literature on its outcomes and implementation issues*. Academic Medicine. 68(1), 52-81.

Anderson, A.S. (1991). *Conversion to problem-based learning in 15 months. in the challenge of problem-based* learning. Boud D and Feletti G. Eds. Kogan Page, London. 1991.

Apache Group, (2002). http://httpd.apache.org/ (last accessed 20/1/02).

Armstrong, R., Freitag, D., Joachims, T. and Mitchel T. (1995) *WebWatcher: A learning apprentice for the World Wide Web*. AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments, Stanford, CA, http://www.isi.edu/sims/knoblock/sss95/mitchell.ps (last accessed 1/12/01).

Atkinson, L. (2001). *Installations: Fast and Easy.* http://www.zend.com/zend/trick/installations.php (last accessed 10/1/02).

Bakken, S. S. (2000). *Introduce to PHP.* http://www.zend.com/zend/art/intro.php#Heading4 (last accessed 12/1/02).

Barrows, H.S. (1986). *A Taxonomy of Problem-based Learning Methods.* Medical Education, 20, 481-6

Barrows, H.S. (1998). *The essentials of problem-based learning*. Journal of Dental Education. 62:9, 630-633.

Barrows, H.S. and Myers, A.C. (1993). *Problem-based learning in secondary school.* Monograph for problem-based learning institute, Ventures in learning Inc. Southern Illinois University School of Medicine, Langhier High School PBL Laboratory School, and Springfield School District 186, Springfield, IL.

Beaumont, I. (1994), *User modeling in the interactive anatomy tutoring system ANATOM-TUTOR.* User Models and User Adapted Interaction 4 (1), 21-45.

Bednar, A. K., Cunningham, D., Duffy, T. M. and Perry, J. D. (1991). *Theory into practice: How do we link?* In G. J. Anglin (Ed.), Instructional technology: Past, present, and future. Englewood, CO: Libraries Unlimited.

Berkson L. (1993) *Problem-based learning: Have the expectations been met?* Acad Med 1993; 68: S79-S88.

Blumberg, P. and Michael, J. (1992) *Development of self-directed learning behaviours in a partially teacher-directed problem-based learning curriculum.* Teach Learn Med 1992; 4: 3-8.

Boud, D. and Feletti, G. (Eds.) (1991). *The challenge of problem based learning.* New York: St. Martin's Press.

Boyle, C. and Encarnacion, A. O. (1994), *MetaDoc: an adaptive hypertext reading system.* User Models and User Adapted Interaction 4 (1), 1-19.

Brown, J. S., Collins, A. and Duguid, P. (1989). *Situated cognition and the culture of learning.* Educational Researcher, 18(1), 32-42.

Browne, D. P., Totterdell, P. A. and Norman, M. A. (1990). *Adaptive user interfaces.* London: Academic Press.

Brusilovsky, P. L. (1992) *Intelligent Tutor, Environment and Manual for Introductory Programming.* Educational and Training Technology International 29 (1), 26-34.

Brusilovsky, P. (1994). *the construction and application of student models in intelligent tutoring systems.* Journal of Computer and System Sciences International, 32(1), 70-89.

Brisilovsky, P. (2000). *Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies.* http://www-aml.cs.umass.edu/~stern/webits/itsworkshop/Brusilovsky.html

Calvi, L. and De Bra, P. (1997) *Using dynamic hypertext to create multi-purpose textbooks.* In: T. Müldner and T. C. Reeves (eds.) Proceedings of ED-MEDIA/ED-TELECOM'97, Calgary, Canada, June 14-19, 1997, AACE, pp. 130-135.

Clancey, W. J. (1986). *Review of Winograd and Flores' understanding computers and cognition: A favorable interpretation.* (STAN-CS-87-1173) Palo Alto, CA: Department of Computer Science, Stanford University.

Davis, P. (1999). *A Brief Introduction to PHP: Hypertext Preprocessor.* http://www.webguys.com/pdavis/Programs/What_Is_PHP/ (last accessed 18/1/02).

Des Marchais, J.E. (1993). *A student-centred, problem-based curriculum: 5 years' experience.* Can Med Assoc J 1993; 148: 1567-1572.

Des Marchais, J. E., Bureau, M. A., Dumais, B. and Pigeons, G. (1992). *From traditional to problem-based learning: A case report of complete curricular reform.* Medical Education, 26: 190-199,1992.

Devitt, P. (1995). *Application of a problem-oriented learning package in medical education.* http://www.ascilite.org.au/conferences/melbourne95/smtu/papers/devitt.pdf

Diego, J., Rivera, Z., Jim, E. and Cooke, J. (2000). *An XML-Based Tool for Building andusing Conceptual Maps in Education and Training Environments.* In: Proceedings for "International Conference on Computer in Education/International Conference on Computer Assisted Instruction 2000" (ICCE/ICCAI 2000), Tai Pei, Taiwan.

Duffy, T.M. (1994). *Corporate and community education: achieving success in the information society.* Unpublished paper. Bloomington, In: Indiana Univeristy.

Duffy, T. M., and Jonassen, D. (1991). *Constructivism: New implications for instructional technology?* Educational Technology, 31(5), 3-12.

Duffy, T.M., Lowyck, J. and Jonassen, D. (1993). *Designing environments for constructivist learning.* Heidelberg: Springer-Verlag.

Gijselaers, W.H. (1996) *Connecting Problem-Based Practices with Educational Theory.* In Wilkerson, L. & Gijselaers, W.H. (eds.), Bringing Problem-Based Learning to Higher Education: Theory and Practice. San Francisco: Jossey-Bass, 1996.

Gonschorek, M. and Herzog, C. (1995). *Using hypertext for an adaptive helpsystem in an intelligent tutoring system.* AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC, 274-281.

Han, B. (2001). *Student Modelling and Adaptivity in Web-based Learning System.* Massey University, Palmerston North, New Zealand.

Höök, K. (1997). *Evaluating the utility and usability of an adaptive hypermedia system.* In: J. Moore, E. Edmonds and A. Puerta (eds.) Proceedings of 1997 International Conference on Intelligent User Interfaces, Orlando, Florida, January 7-9, 1997, ACM, 179-186.

Hulse, S. H., Egeth, H., and Deese, J. (1980). *The psychology of learning* (5th ed.). New York: McGraw-Hill.

Johnson, W. L., and Shaw, E. (1997). *Using agents to overcome deficiencies in web-based courseware.* Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web", 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August 1997.

Jonassen, D. H. (1991). *Evaluating constructivistic learning.* Educational Technology, 31(9), 28-33.

Kantrowitz, M. P., Kaufman, A., Mennin, S., Fulop, T. and Guilbert, J. (1987) *An experimental approach to change relevant to health needs.* Innovative Tracks at Established Institutions for the Education of Health Personnel. Geneva: World Health Organization (WHO) Offset Publication 0303-7878: 101, 1987.

Kaufman, A. (1985). *Lessons from Successful Innovations: Implementing Problem-Based Medical Education*. Springer Publishing Co., New York. 1985.
Keller, J. M. (1979). *Motivation and instructional design: A theoretical perspective.* Journal of Instructional Development, 2(4), 26-34.

Koschrnann, T.D., Feltovich, P.S., Myers, A.M. and Barrows, H. S (1990). *Designing communication protocols for a computer-mediated tutorial laboratory for problem-based learning.* Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care, Los Alamitos, CA: IEEE Computer Society Press, 1990, 464-468.

Lebow , D. (1993). *Constructivist values for systems design: five principles toward a new mindset*, ETR&D, Vol.41, 4-16.

Mathé, N. and Chen, J. (1996). *User-centered indexing for adaptive information access*. User Models and User Adapted Interaction 6.

Meloni, J. (2001) *Are you being served? An introduction to PHP.* http://www.zdnet.com/devhead/stories/articles/0,4413,2576930,00.html (last accessed 8/1/02).

Merral, G (1999). *PHP/MySQL tutorial.*
http://hotwired.lycos.com/webmonkey/programming/php/tutorials/tutorial4.html)
(last accessed 8/1/02).

Merrill, M. D., Kowalis, T. and Wilson, B. G. (1981). *Instructional design in transition.* In F. H. Farley, & N. J. Gordon (Eds.), Psychology and education: The state of the union (pp. 298-348). Berkeley: McCutchan.

Milter, R. G. and Stinson, J. E. (1993). *Educating leaders for the new competitive environment.* In Gijselaers, G., Tempelaar, S., Keizer S. et al (Eds.), Educational innovation in economics and business administration: the case of problem-based learning. London: Kluwer Academic Publishers.

MySQL AB Company (2002). *MySQL*
http://www.mysql.com (last accessed 10/1/02).

Neufeld, V. R. and Barrows, H. S. (1974). *The "McMaster philosophy": An approach to medical education.* Journal of Medical Education , 49(11): 1040-50, 1974.

Newby, T. J., Stepich, D. A., Lehman, J. D. and Russell, J. D. (1996). *Instructional technology for teaching and learning: designing instruction, integrating computers, and using media,* NJ: Prentice Hall.

Nielsen, J. (1990). *Hypertext and hypermedia.* San Diego, Academic Press

Nikov, A. and Pohl, W. (1999). *Combining user and user modeling for user-adaptivity systems.* Human Computer Interaction – Ergonomics and User Interaces (Eds. H.-J. Bullinger & J. Ziegler).

Oppermann, R., Rashev, R. and Kinshuk. (1997). *Adaptability and adaptivity in learning systems.* Knowledge Transfer (volume II) (Ed. A. Behrooz), pAce, London, 173-179.

Pennell, R. and Deane, M. E. (1995). *Web browser support for problem-based learning.*
http://cadfl.uws.edu.au/ascilite95/lcipaper.html (last accessed 16/11/01).

Perkins, D. N. (1991). *Technology meets constructivism: Do they make a marriage?* Educational Technology, 31(5), 18-23.

PHP Group, (2002). *PHP installation.*
http://www.php.net (last accessed 6/1/02).

Richey, R. D. (1986). *The theoretical and conceptual bases of instructional design.* New York: Nichols.

Savory, J. R., and Duffy, T. M. (1995). *Problem based learning: an instructional model and its constructivist framework.* Educational Technology, 35, 31-38.

Schunk, D. H. (1991). *Learning theories: An educational perspective.* New York: Macmillan.

Self, J. A. (1994). *Formal approach to student modeling.* Student Modeling: the Key to Individualized Knowledge-Based Instruction. (Ed. McCalla,G.I., & Greer, J.E.), Spring-Verleg, Berlin.

Shin, J. H., Haynes, R. B. and Johnson, M. E. (1993) *The effect of problem-based, self-directed undergraduate education on lifelong learning.* Can Med Assoc J 1993; 148: 969-976.

Snelbecker, G. E. (1983). *Learning theory, instructional theory, and psychoeducational design.* New York: McGraw-Hill.

Spaulding, W. B. (1991). *Revitalizing medical education.* BC Decker Inc., Philadelphia, Hamilton (Canada). 1991.

Spiro, R. J., Feltovich, P. J., Jacobson, M. J. and Coulson, R. L. (1991). *Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains.* Educational Technology, 31(5), 24-33.

Supercuts. (2002). *Supercuts*
http://www.supercuts.com (last accessed 6/1/02).

Vassileva, J. (1995). *Dynamic courseware generation: at the cross point of CAL, ITS and Authoring.* Proceedings of the AI-ED Workshop on Authoring Shells for Intelligent Tutoring Systems.

Vassileva, J. (1996). *A task-centered approach for user modeling in a hypermedia office documentation system.* User Models and User Adapted Interaction 6 (this issue).

Warries, E. (1990). *Theory and the systematic design of instruction.* In S. Dijkstra, B. van Hout Wolters, & P. C. van der Sijde, (Eds.), Research on instruction: Design and effects (pp. 1-19). Englewood Cliffs, NJ: Educational Technology.

Winn, W. (1991). *The assumptions of constructivism and instructional design.* Educational Technology, 31(9), 38-40.

Woods, D. (1994). *Problem based earning: How to get the most from PBL.* McMaster University, 1994.

Woolfolk, A. E. (1993). *Educational psychology.* Bosten: Allyn and Bacon.