

WXES 3182

**Computer Aided Learning
Environment for Biology**

Prepared by:

**FOONG PEK HIN
WEK 98023**

Under the supervision of
Dr. Roziati Zainuddin

Moderated by
Dr. Syed Malek Fakar Duani

Faculty of Computer Science and Information Technology
University of Malaya

Dissertation submitted in partial fulfillment of the requirement for the Degree of
Bachelor of Computer Science.

Session 2000/2001

Date of Submission:
12th September 2001

ABSTRACT

Traditional learning on biology requires more of memory rather than understanding. As better educational technology emerges, we are able to utilise the computer as an important tool in learning. Many educational and pedagogical issues, particularly in the field of Artificial Intelligence are looked at to help consider a good design for a computer learning tool. Existing computer aided learning materials are also studied to pinpoint their better features and incorporate them into the learning tool.

A learning tool is proposed for effective learning in relating facts, rules, text and graphics. This system will have a Study Section containing notes that incorporates multimedia features to enhance motivation and interest of students. Apart from that, it will have an intelligent Database Section where students are given the opportunity to improve and test their knowledge. Not only that, this section will see the interaction of students with the program as they are allowed to add and remove information from the program.

This learning tool would be developed following the Rapid Application Development (RAD) methodology. It will be programmed using Visual Prolog 5.1. The use of Visual Prolog 5.1 will help to make a presentable GUI as well as an intelligent system. As for the system requirements of the finished product, it would be able to run on a stand-alone computer with a CD-ROM/floppy drive and a generic hard disk.

ACKNOWLEDGEMENTS

Many people have helped me in the preparation of this report. First of all, I would like to thank my supervisor, Dr. Roziati Zainuddin. She has given me guidelines in the research process, how to conduct the research and how to write the report. Most of the time, she has pointed out errors or inconsistencies in my effort to complete the report. I would also like to thank her for her faith in me and also for helping me when I needed her the most.

Besides that, I would also like to thank my moderator, Dr. Syed Malek who provided valuable comments, many of which have been incorporated into the report. I appreciate his support and encouragement.

Lastly, I would also like to thank my family and friends who have encouraged me throughout the process of writing this report. I thank them for their comments on my work because without them, I would not have gotten the chance to improve myself.

Thank You.

TABLE OF CONTENTS

Abstract.....	i
Acknowledgements.....	ii
Table of Contents.....	iii
List of Figures.....	vii
List of Tables.....	vii
1.0 Introduction	
1.1 Background Study.....	1
1.2 Objectives.....	2
1.3 Project Definition.....	3
1.4 Project Scope.....	4
1.5 Methodology.....	6
1.6 Activity Schedule.....	10
2.0 Literature Review	
2.1 The Role of Technology in Education.....	11
2.1.1 Computers and Knowledge Transfer.....	12
2.1.2 Computers and Learning through Exploration.....	12
2.1.3 Computers and the Intellectual and Emotional Development of the Learner.....	13
2.1.4 The Promise of Educational Technology.....	14
2.2 The Psychology of Learning.....	16
2.3 Computer Aided Learning (CAL).....	18
2.4 Intelligent Computer Aided Learning (ICAL).....	21
2.5 Existing Intelligent Computer Aided Learning System	23
2.5.1 Computer Aided Learning in Mathematics (CALM)	23

2.5.2 Knowledge Based Computer Instruction System (KBCIS)	26
3.0 System Analysis	
3.1 Fact Finding Techniques.....	29
3.2 Requirements Analysis.....	30
3.2.1 Functional Requirement.....	30
3.2.2 Non-Functional Requirement.....	31
3.3 Selection of Programming Languages.....	33
3.3.1 Macromedia Authorware 4.0.....	33
3.3.2 Visual Prolog.....	34
3.4 System Requirement.....	37
3.4.1 Hardware.....	37
3.4.2 Software.....	37
4.0 System Design	
4.1 Hierarchy of the ICAL contents.....	38
4.1.1 Study Section.....	39
4.1.2 Database Section.....	39
4.2 Knowledge Database Design.....	40
4.2.1 Classification Module	40
4.2.2 Data Module.....	40
4.3 Interface Design.....	41
5.0 System Implementation and Testing	
5.1 Development Environment.....	43
5.1.1 Hardware.....	43
5.1.2 Software.....	44
5.1.3 Software Tools Used to Produce Report.....	44

5.2 Program Development.....	45
5.2.1 Reviewing the Program Documentation.....	45
5.2.2 Designing the Program.....	45
5.2.3 Coding the Program.....	45
5.2.4 Testing the Program.....	46
5.2.5 Document the Program.....	46
5.3 Program Coding.....	47
5.3.1 Coding Approach.....	47
5.3.2 Coding for the Study Section.....	49
5.3.3 Coding for the Database Section.....	51
5.4 System Testing.....	54
6.0 System Evaluation and Conclusion	
6.1 Problems Encountered and Their Solutions.....	55
6.1.1 Difficulty in Choosing a Suitable Programming Language.....	55
6.1.2 Inexperience Using the Programming Language.....	56
6.1.3 Determining the Scope of the System.....	56
6.1.4 My Level of Knowledge of the Subject.....	57
6.2 System Strengths.....	58
6.2.1 Friendly User Interface.....	58
6.2.2 Ease of Expansion.....	58
6.2.3 System Has a Dynamic Database.....	59
6.2.4 Interactive Environment.....	59
6.3 System Limitations and Future Enhancements.....	60
6.3.1 Insufficient Information Provided by the System.....	60
6.3.2 Requires Exact Matching.....	60

6.3.3 Future Enhancements.....	60
6.4 Conclusion.....	61
References.....	63
User's Manual.....	65

University of Malaya

LIST OF FIGURES

Figure 1-1.....	7
Figure 2-1.....	19
Figure 3-1.....	36
Figure 4-1.....	38
Figure 4-2.....	42

LIST OF TABLES

Table 1-1.....	10
----------------	----

1.0 INTRODUCTION

1.1 Background Study

The problems with the conventional teaching approach to large classes, which relies mostly on the text books and the questions of the review exercises at the end of each chapter, are well known. Apart from the fact that face-to-face consultation time between students and lecturers can never be sufficient, the amount of time required to mark a large number of assignments has usually prevented a lecturer from providing detailed feedback.

In the past, the only way to reduce these problems is through providing more teaching assistants, which is generally very costly and has proved ineffective in many cases. Recently, the promotion of the Computer Assisted Learning (CAL) has targeted these areas. However, there were problems that came with CAL such as its difficulty to provide meaningful feedback to guide students in problem solving processes.

The development of the Intelligent Computer Aided Learning (ICAL) is an attempt to further advance the capabilities of CAL by applying various techniques in Artificial Intelligent (AI). Specifically, ICAL adopts the "learning-by-doing" as the basic instructional approach. In this approach, students are required to engage activity in the instructional process to formulate and test their ideas. A typical ICAL package contains 3 components – the knowledge of domain, the knowledge of the learner (student) and the knowledge teacher strategies.

1.2 Objectives

In order to fulfill the requirements of students in having a complementary tool to do their own revision and improve their learning instead of merely reading the text book or notes delivered by the traditional instructor, the software developed must be able to act as a supporting tool to foster effective learning by guiding and coaching them.

There are several objectives that we will address in the development of this system:

- To fully utilise the potential of Information Technology to improve learning.
- To use AI techniques in the ICAL package to provide an effective and efficient tool for students to study Biology.
- To encourage computer literacy among students.
- To prove that computers can be used beneficially as an accessory in the process of learning.

1.3 Project Definition

When we talk about education, the first thing that comes to mind is schools, colleges, universities or some sort of physical infrastructure where the process of learning and teaching takes place. Specifically for the subject of Biology, the conventional method of learning entails going to lectures where the lecturer will disseminate information in chunks. At this point, the student is expected to capture the essence of the lecture. After that, the student must read revision and textbooks in order to pick up more facts. There may be lab periods where the student can obtain practical experiences. However, since the theory and practical studies are usually held on separate sessions, the student will find it hard to relate the facts. In the end, the student will resort to memorizing facts in order to pass the subject. There will be no real understanding on the part of the student.

This proposed system would attempt to address these problems. The utilisation of educational technology offers the opportunity and potential of restructuring the educational system to extend the learning environment beyond the four walls of the classroom.

- The student would be able to learn at a time and place that suits him because the software is always available, unlike a lecturer.
- The intelligent features of the system would help to guide the student towards understanding the subject.
- Multimedia features would be incorporated to enhance motivation and understanding.

1.4 Project Scope

This project will focus on the development of an effective tool for learning Biology. The application will be localised on a particular Biology sub-topic. This is because it would be impossible to cover the entire field of Biology. The project will include the development of a system comprised of two main components. They are the Study Section and the Database Section.

The Study Section will contain information, mostly in the form of text accompanied with charts, pictures and explanatory graphics. Some of the terminologies of Biology might also be hyperlinked to their definitions to ease understanding of the subject. Navigation will be non-linear and the user will be able to browse through the entire contents at their leisure. The Database Section will contain a collection of facts about the particular sub-topic of Biology we wish to talk about. In this section, users will be able to test out their understanding and knowledge of the topic. In fact, this section can be expanded and its database enriched by accepting inputs from the user, therefore making it a dynamic tool for learning.

This system will act as a learning companion for students. However, because of the nature of this learning tool, it will be important to provide an in-depth view of the subject of Biology. As a result, the system will be very focused, must be custom-made and will not be useful if applied in other fields.

The final product will be stored on a CD and can be used on any stand-alone computer with a CD-ROM drive. Alternatively, it can also be stored on floppy diskettes and used on any stand-alone computer with a floppy disk drive. However,

this system will not be able to interact with the Internet nor will it include networking support for use in a networked environment. It is mainly a stand-alone system.

University of Malaya

1.5 Methodology

This project will be developed using the Rapid Application Development (RAD) methodology.

The RAD methodology is a very flexible methodology. It incorporates prototyping and user feedback as its main mechanisms. It is usually chosen in cases where a large user community will have significant input to the system, the requirements of the new system are unclear, or there is a high degree of possibility that the requirements and feature set will change as the project proceeds.

RAD is the process of creating a new software system by involving the user community in all phases of the system creation. It implies a user driven design. Deliveries of the system are broken into several milestones, each containing more system components than the previous milestone delivery. The first delivery is usually a prototype of the complete system to include system flow from form to form to report, etc. All appropriate users have a chance to review each delivery before work commences on the remainder of the system. User input to changes is a core requirement to the success of this methodology. RAD facilitates early user acceptance, and ensures that all important users of the system have input to system functionality before its final delivery. RAD development is an important tool in the risk management of a custom software system.

RAD is a manifestation of Vilfred Pareto's law known as the 80 - 20 Rule. It states that 80% of XX is caused by 20% of YY. More specifically, 80% of the costs of a complete system are due to 20% of the features included in the system. Conversely, 20% of the system costs are due to 80% of the system features. RAD methodology allows the most important 80% of the system features to rise to the top of the list and be integrated into the system - at a cost of 20% of the complete system.

The graphic below illustrates a typical chain of RAD process events. Note that the left side dotted line from 'Object Creation' through 'Return to Brainstorm, Requirements, Analysis' is optional. If the system designers are well versed on the system requirements then the first RAD deliveries may not have to be redesigned.

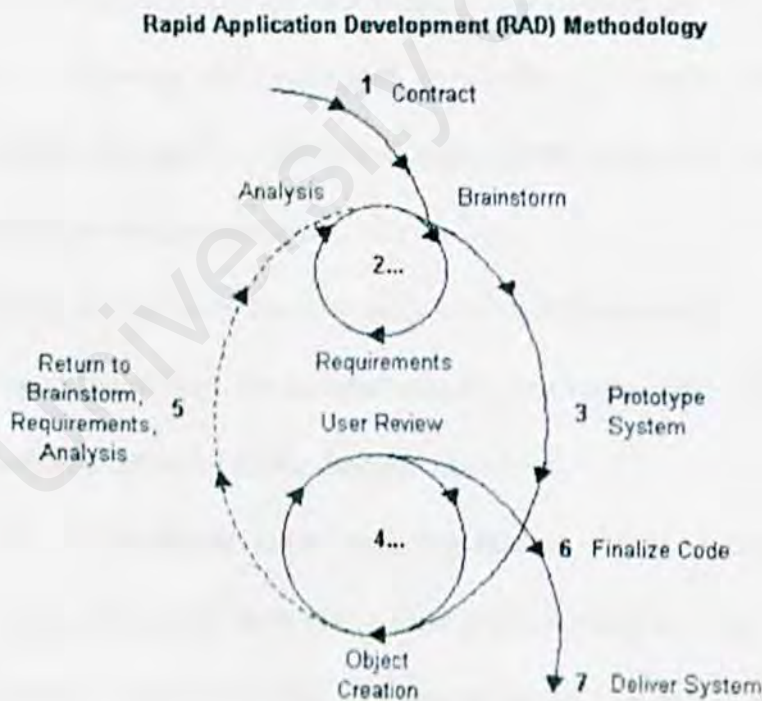


Figure1-1: A diagram illustrating the RAD methodology

The following further describes the RAD Methodology:

1. Once the project development starts, the **Brainstorm**, **Requirements**, and **Analysis** steps may be repeated until enough **information is known** to create a prototype.
2. The prototypes are mock-ups at this stage, and may be completely thrown out if necessary. Code is not deliverable quality. The prototypes will usually change significantly after user feedback, and the impact of these changes are minimized if the prototypes are bare bones - for visual and navigational demonstration only.
3. The Object Creation and User Review steps can be repeated multiple times until the system is correct and accepted by the user community. Objects might include data entry forms, high profile management reports, engineer reports, etc. In order to meet tighter schedules, *it is important for the group of users that are reviewing the prototypes to commit to timely feedback*. In this methodology, the project schedule is often driven by the time taken by the user community to review deliveries.
4. Optionally, if the User Review step results in fundamental changes in the direction of the system, the **Brainstorm**, **Requirements**, and **Analysis** steps can be repeated as depicted in the diagram above.
5. Once all of the above steps are completed to the acceptance of the user community, the system code is finalized in preparation for final delivery.
6. The system is delivered (with all documentation), installed, and configured. Users are trained, and the system moves into maintenance mode.

Advantages of the RAD methodology:

1. Flexible and adaptable to changes.
2. Prototyping applications gives users a tangible description from which to judge whether critical system requirements are being met by the system. Report output can be compared with existing reports. Data entry forms can be reviewed for completeness of all fields, navigation, data access (drop down lists, checkboxes, radio buttons, etc.).
3. RAD generally incorporates short development cycles - users see the RAD product quickly.
4. RAD involves user participation thereby increasing chances of early user community acceptance.
5. RAD realizes an overall reduction in project risk.

Disadvantages of RAD methodology:

1. Unknown cost of product. As mentioned above, this problem can be alleviated by the customer agreeing to a limited amount of rework in the RAD process.
2. It may be difficult for many important users to commit the time required for success of the RAD process.

1.6 Activity Schedule

The activity has been planned and the scheduling will be based on the few sections of project development: Analysis, Design, Implementation, Testing and Documentation.

The following table shows the activity schedule from the beginning of the project until the Design phase. Note that the Implementation and Testing phase has not been carried out yet at this moment. As for the Documentation phase, it is planned that it will continue along the entire length of the project.

	June	July	Aug	Sept	Oct	Nov	Dec	Jan
Analysis								
Design								
Implementation								
Testing								
Documentation								

Table 1-1: Activity Schedule

2.0 LITERATURE REVIEW

2.1 The Role of Technology in Education

One of the earliest insights into the educational applications of technology was that interactive computer-based systems admit the possibility of individualizing the educational process to accommodate the needs, interests, current knowledge, and learning styles of each particular student. Even the earliest drill-and-practice based computer-assisted instruction systems, in which the student was exposed to successive blocks of textual material and answered a series of questions posed by the computer, typically offered the advantages of self-paced instruction.

In recent years, however, many researchers have begun to focus on the potential of technology to support certain fundamental changes in the pedagogic models underlying our traditional approach to the educational enterprise. Technology, thus, changes the way in which students and teachers interact. When students use technology to identify and collect information, they no longer depend on the teacher as the source of information. Schools, colleges and universities will no longer be the sole distributor of knowledge, and the learning process will now take place without the constraints of time and space. Soon it will be difficult to actually categorise students as full-time or distance learners. As a matter of fact, face-to-face learning may even become a supplementary activity in the future.

2.1.1 Computers and Knowledge Transfer

The advocates of the transfer theory believe in the importance of a well-structured and organised preparation and transferral of learning materials for a teaching process in which the learner takes up the role of the receiver of knowledge. Drill and practice programs and intelligent computer aided learning systems can support well-organised and structured knowledge delivery. They can provide a teaching situation in which the learner assumes a receiving role. Complete guidance is given by the software, and teacher interference is only required should the learner have difficulties with the functionality of the software. Additionally, these applications can generally provide a more stimulating and attractive context in which knowledge can be presented and exercises carried out than, for example, a book of exercises.

2.1.2 Computers and Learning through Exploration

Within the theory of teaching as an exploratory journey through the subjects under study, the learner is expected to have quite some influence on the kind of learning process, the pace at which learning takes place and on the subject area that is being studied.

There are numerous software applications that provide the learner with an environment in which he can adopt the role of an explorer (Scrimshaw 1997). Encyclopaedias on CD-ROMs and adventure games, for example, encourage students to explore. Within these applications the learner has control over the nature of and the order in which the subject material is being accessed. Games have the additional advantage that they tend to motivate students. Productivity tools, such as word processors and graphics packages, offer opportunities for teachers to set up specific

'landscapes' in which the student can then be encouraged to explore certain issues or concepts (Scrimshaw 1997). The teacher may, for example, present his students with a pre-filled spreadsheet or database for the student to explore.

At the same time these kinds of applications have the inherent capability to adjust themselves to their users. Users can choose the level of complexity of their input they feel comfortable with, thereby reducing the risk of frustration and turning learning into a motivating experience.

2.1.3 Computers and the Intellectual and Emotional Development of the Learner

Following the growing theory means nurturing a learner's intellectual and emotional development. The subject under study is only relevant in regards to the impact it has on the personal growth of the learner.

Again, productivity tools, such as word processors and graphics packages, can provide a suitable environment. The learner can be left to be creative in words and pictures in an unstructured and spontaneous manner (Scrimshaw, 1997). Similarly, programming languages require the user to be creative and express his conception of a process. Databases allow for the storing and retrieving of data in structures that are meaningful to the individual.

2.1.4 The Promise of Educational Technology

The following are a few of the better-known examples of the successful application of technology to education in the United States of America (as reported in the "Report to the President on the Use of Technology to Strengthen K-12 Education in the United States, March 1997):

- *Blackstock Junior High School (California)*: This school has ten "smart classrooms," including one in which students can use computer-aided design (CAD) software to describe products that are then fabricated using a computer-controlled flexible manufacturing system. Higher test scores and improvements in comprehension, motivation, and attitude have been reported for the predominantly Hispanic student body.
- *Carrollton City School District (Georgia)*: Computer technology is used in this school district as part of a novel program that has succeeded in reducing the dropout rate from 19 percent to 5 percent, and the failure rate in ninth grade algebra from 38 percent to 3 percent.
- *Carter Lawrence School (Tennessee)*: Students in selected classrooms within this Nashville middle school used technology in various ways as part of a program called Schools for Thought, which is based largely on constructivist principles. Sixth-grade SFT participants scored higher on a number of components of Tennessee's mandated standardized achievement test than students in matched comparison classrooms, and demonstrated substantially stronger critical thinking skills in complex performance assessments involving high-level reading and writing tasks. Absenteeism and student withdrawal rates were also dramatically lower among SFT students.

- *Christopher Columbus Middle School (New Jersey)*: Perhaps the most widely publicised example of the successful application of educational technology, this inner-city school in Union City implemented a reform program that (along with other important changes) provided all seventh-grade students and teachers with access to computers and the Internet, both at school and at home. The performance of its 91 percent Hispanic student population, the majority economically disadvantaged, improved from significantly below to somewhat above the statewide average in reading, language arts, and math.

2.2 The Psychology of Learning

Learning is any relatively permanent change in behaviour (or behaviour potential) produced by experience and not caused by physical maturation or growth (Baron, 1989).

Jennings (1993) lists four fundamental sources of knowledge:

- Believing (or Authoritative Knowledge)
- Thinking (or Rational Knowledge)
- Sensing (or Empirical Knowledge)
- Feeling (or Intuitive Knowledge)

These four sources seem to interplay with each other in the learning process to build knowledge. We can either start from scratch and use a rational logic to come to a conclusion or we can accept an authoritative conclusion. Further these could be modified via new intuitive or empirical 'evidence'. Many of the questions of learning theory revolve around whether learning methods can be isolated, presumably as no teaching method can incorporate all at once.

Regardless, cognitive psychologists have determined that understanding knowledge is the best way to learn. This way, the information will be semantically encoded into the brain, thus prolonging the time it takes for the information to dissipate from memory. Besides that, we also learned that relating a piece of information to another piece of information that may be known enhances memory results, as suggested by the *elaboration hypothesis* (Craik and Tulving, 1975).

Learning may also take place in the form of *guided discovery*, which is one of the richest modes of learning. For example, a student *who is trying to answer* a set of questions may inadvertently makes mistakes. If *guidance is given during* this process, the student can discover new information previously *not known to him*. As the phrase '*we learn through mistakes*' suggests, the student will be able to remember the facts related to the incident much better and learning will be more effective.

Further, we gain knowledge in different phases. We don't, except for very trivial tasks, become experts overnight. Jonassen (1992:387) describes three such phases:

- *Introductory Knowledge* acquisition, epitomised by the naive learner with no prior subject knowledge;
- *Advanced Knowledge* acquisition which must be attained to solve complex domain or context-dependent problems and;
- *Expertise*, where experts have an interconnected knowledge structure. All of these require different forms of instruction.

As a whole, all these issues of learning should be taken into consideration when designing learning tools.

2.3 Computer Aided Learning (CAL)

CAL is the term used to describe a system for learning that is usually interactive, self-paced and comprehensive which offers instruction and teaching. It is an educational medium in which instructional content or activities are delivered by a computer. It is equivalent to Computer Assisted Instruction, Computer Aided Instruction (CAI), Computer Based Instruction (CBI) and Computer Based Learning (CBL).

Most CAL software are designed linearly and incorporates the use of behavioural modifications such as:

- Applying appropriate knowledge reinforcers such as text, audio or video.
- Shaping, chaining, modeling, punishments and rewards.
- Scoring and/or monitoring systems.
- Status of progress.

CAL comes in various forms such as Drill and Practice, Simulations, Tutorial and Game.

Figure 2-1 : Routing within a CAL system.

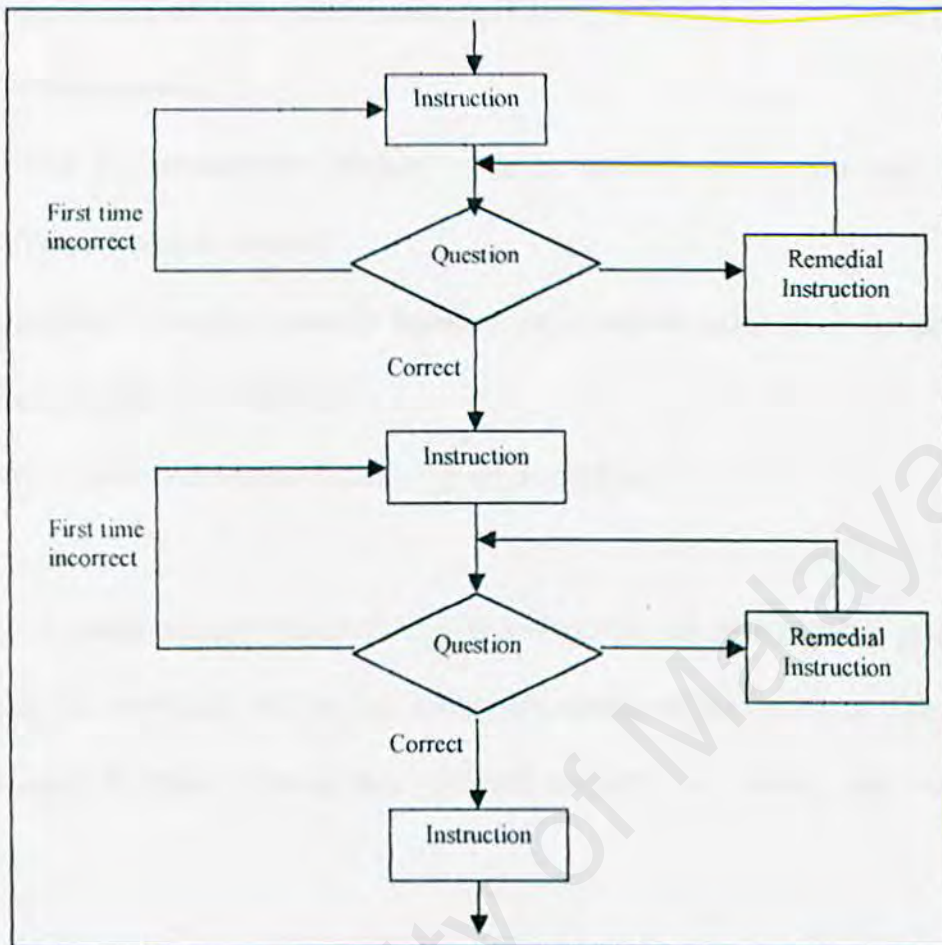


Figure 2-1 illustrates the routing within a CAL system. The information will be presented to the user after a series of questions are asked. If the correct answers are detected, the next phase of instruction is entered. If the incorrect answers are detected, the instructional material may be presented in a different format. If, after the material has been repeated, the student again answers incorrectly, then remedial instructions may be presented.

CAL provides a number of benefits:

- Appropriate use of CAL can release staff time, allowing greater investment in small group teaching.
- Interactive and multimedia delivery systems enhance motivation and, if used effectively, reinforce learning.
- Cal provides a student-centered learning environment tailored to the pace and learning style of the individual.
- Student receives immediate feedback from the system.

However, the disadvantage of these CAL systems include not being able to provide an explanation that will help resolve the misunderstanding of the problem. The system should generate a certain response to a particular situation, as a human tutor would be able to do.

2.4 Intelligent Computer Aided Learning (ICAL)

Intelligent CAL is similar to a CAL system but more intelligent in giving guidelines and tutoring. While the traditional CAL was mainly developed by educational researchers, who tried to solve their practical problems through non-AI techniques, ICAL was initiated by computer scientists, who tried to explore the capability of AI techniques in the process of learning and teaching. Specifically, ICAL adopts the “learning-by-doing” as the basic instructional approach (Sleeman & Brown, 1982).

A typical ICAL system contains most or all of the following: a domain expert, a teaching expert, a diagnostic expert and a student model. A domain expert provides the knowledge of both procedural and factual that students need to learn. The diagnostic expert uses rules to analyse student responses. The teaching expert determines the strategy for teaching the student based on the current state of the student model. Through the interaction of these models, ICALs are able to make judgements about what the student knows, and how well the student is performing without the intervention of a human instructor.

Mandl and Lesgold (1988) advances the concepts further by emphasising the importance of providing flexible and adaptive user interaction, providing different viewpoints for users to access information, delivery of contents according to users' knowledge and skill level, and providing updated assistance to students according to user's learning progress.

Students can use the intelligent tutorial to evaluate their understanding of the subject, reviewing only those sections that need additional study. The hints and

guidelines for solving the problems will direct and help the students master the particular topics.

University of Malaya

2.5 Existing Computer Aided Learning Systems

2.5.1 Computer Aided Learning in Mathematics (CALM)

The CALM Project for Computer Aided Learning in Mathematics started in the Department of Mathematics at Heriot-Watt University in Edinburgh in 1985. Heriot-Watt University is a technological university with large groups of engineering, science and business undergraduates on its Riccarton campus. CALM has delivered successful and innovative teaching and learning to large groups of undergraduates in each academic year since 1986.

Calculus is an ideal medium to bring applications of mathematics to life. The 22 units of CALM courseware produced cover the syllabus of a typical course on calculus with differentiation, integration, an introduction to numerical methods and elements of ordinary differential equations. Each unit includes the topics first encountered in approximately two lectures of the course. Mathematical modelling and the development of mathematics in Engineering and Applied Physics is an important feature of CALM.

For example:

1. *State the Rate* invites the students to work through a problem involving the filling of a cup from a coffee dispenser encouraging design considerations;
2. *Fireman* is a model of the trajectory of water from a hose which properly directed puts out a fire; and
3. *Escape from Colditz* asks the students to work with calculus and numerical methods to solve an optimisation problem.

Each unit has been constructed around:

- Theory sections, to consolidate the conventional lecture;
- Worked example sections, to teach the methods of calculus;
- Motivating examples, including mathematical "games", experimental elements and mathematical models; and
- Test sections, to enable students to assess their own strengths and weaknesses and to allow the teacher to monitor individual progress.

The units are designed to allow the students complete control over their route through the tutorial. The course for which these units are part, is mainly assessed in a conventional way. The test section of each unit can be taken at any one of three levels to cater for a spectrum of ability. The test offers different types of help at each level. Students can start a topic at the easiest level and progress to the hardest level as they gain in confidence.

The students' marks and test answers are recorded so that the teacher can monitor their progress. Students who are working well are sent encouraging messages. Those who are in difficulty are detected early in the year and given extra attention. By viewing the recorded answers, the teacher is able to identify the source of a student's problem and send an appropriate message.

Funding from the Department of Trade and Industry and a number of leading British firms led to an extension of the original CALM Project in 1990. Courseware has been authored at the level of Scottish Higher in Mathematics which is within core A Level Mathematics thus providing some pre-university material. It includes

trigonometry, basic algebra, analytic geometry, differentiation, and elementary probability. This new courseware builds on the experience gained in the earlier project and contains a number of extra features. Using the versatile authoring language Authorware the topics are covered with:

- a new section (in each topic) of self-assessment questions which are designed to provide rapid feedback to right and wrong answers;
- linking between appropriate theory, worked examples and self-assessment sections;
- enhanced graphical displays and more animated sequences;
- an overview of each topic and a better help system; and
- a more flexible unit test.

2.5.2 Knowledge-Based Computer Instruction System (KBCIS)

The Knowledge-Based Computer Instruction System (KBCIS) is developed by Fan, J. P., Mak, T. K. and Shue, L. from the University of Wollongong, Australia. This system was designed as an effective teaching-learning and examinations tool for large Accounting classes. The design of this system incorporates the knowledge-based system approach, which enables this system to store problem solution knowledge and utilise it for marking problems of the same nature but of different settings, and with a great flexibility.

Specifically, this system provides lecturing material, exercise problems, help, hints, and solution procedures, which can be accessed by students according to their ability and the criterion set by the lecturer-in-charge. Based on a representative problem, the system is able to generate similar problems with different settings and parameter values, and mark the answers from students accordingly. Feedback for each problem could be provided if needed. The system can access the student log to monitor the performances of each individual student in a class, and help the lecturer to determine the proficiency status for each topic. This real time analysis also makes it possible for the lecturer to pre-set performance criterion to determine the status of a student, and hence provide appropriate topic and exercise problems.

The principle objective in designing the KBCIS is to enhance and extend the current teaching-learning function of large Accounting subjects, so that the traditional constraint of teaching and learning through personal contact between students, lecturers, or tutors can be minimised. It is envisaged that through this system, the requirements of locality and timing for delivering subject material can be greatly

reduced or even eliminated in some cases. The goals in developing this system encompass three aspects of teaching: lesson delivery, evaluation, and feedback.

Under lesson delivery, it is intended for the system to lessen the dependency of students on formal classroom teaching and tutorial exercises. The present and past lecturing material and tutorial material should be made available to students for retrieval at anytime. In addition, it is vital for the system to be able to automatically generate exercise problems for a given topic, and have the capability to mark the answers properly and instantly. In this way, the system will allow students to practice with a wide range of problems to raise their proficiency and hence alleviate the demands for consultation requests. The specific goals in this area are to: (1) make available lecturing and tutoring material for retrieval by students at anytime, (2) generate exercise questions upon requests from students, and (3) mark answers from students properly and instantly.

Under evaluation, the system must be able to facilitate the development of examination questions with flexibility and efficiency. The types of questions which may be developed include the traditional stand alone questions, and the questions with more than one level of structure. In addition, regardless of the types of questions, the system must be able to mark answers from students with any scheme as is currently used by lecturers. One particular emphasis in this part is the ability for the system to provide partial credit for the portions of a question which are correctly answered. For personal evaluation, the system must be able to determine, just like a lecturer, if a student is good enough to go to the next topic or needs to practice more with the current topic, or needs to go back to an earlier topic. The specific goals for this part

are to: (1) provide a flexible and efficient platform for developing examination questions, (2) provide an evaluation platform for developing flexible marking schemes, and (3) provide a monitoring mechanism to measure progress of students.

Under feedback, the system must be able to explain why an answer is wrong, and indicate the source material for students to refer to, and, if necessary, provide the step-by-step problem solving procedures for those who repeatedly fail the same question. In catering for the needs of each individual student, the system must maintain a complete operation and performance records for every student, which includes the amount of time each student spends for each topic, the number of times a student tries a topic and the marks for each problem. With these, the system can identify those students who are underperforming, or are not participating enough, and notify the lecturer to initiate necessary actions before it becomes too late. The specific goals for this part are to: (1) provide reference information for problems which are wrongly answered, (2) indicate problem solving steps if necessary, and (3) collect performance statistics of students to help lecturers to identify problem areas or individual students who need help.

3.0 SYSTEM ANALYSIS

3.1 Fact Finding Techniques

Information regarding the needs, views and suggestions of possible ICALs are collected for further analysis. Before performing the detail analysis of the requirement for the learning system, facts have been gathered from:

a) Reading

Reading had been done to look for current existing ICALs or its equivalents. Journals, articles and books from the library are referred to for information.

b) The Internet

The Internet is the largest information warehouse in the world. As such, it is used to get information on the same or similar system. Primarily, the Internet played an important role during fact gathering. Searching through the Internet yielded a lot of useful materials.

c) Observation and Experience

In this process, the analyst relates his observations and experiences to the system. My experience as a Biology student helps me to understand the pedagogical issues behind the learning of Biology. In addition, it provides some understanding of the needs of students and what a CAL can do for them.

3.2 Requirements Analysis

This section consists of two sub-sections, which are the functional requirement and non-functional requirement.

3.2.1 Functional Requirement:

This section is focus on the topic of how the ICAL application interacts with its operating environment. The best way to develop and maintain a large program is to construct it from smaller pieces each of which is more manageable than the original program. The technique is called “divide and conquer”.

There are 2 main sections to the proposed learning system:

1. *Study Section*

This section will contain mostly information in the form of text and graphics. This section’s function would be to provide information to the student, much like a lecture would.

2. *Database Section*

The database section is considered the intelligent component of the system. It is here that users will be able to interact with the system. This section can act as a place where users can gauge how much they have actually learnt besides gathering further knowledge of the subject. Also, this section can act as a storehouse of knowledge for the user as they can introduce additional information to the database, and come back later to retrieve that knowledge.

3.2.2 Non-Functional Requirement:

The system development is constrained by non-functional requirements.

a) Portability:

Portability is the capability of a system to operate on different platforms. At the moment, the learning system can only support the Windows platform.

b) Scalability:

The ability of the system to migrate to other machines, which is greater or lesser power without many changes to the system itself.

c) User-friendliness:

Users may find it difficult to operate the system if the design of the interface is not user-friendly enough. Opinions from target users would be invaluable to design a better interface.

d) Flexibility:

This feature is regarding the capability of the system to adopt new technologies and resources as well as implementations in a changing environment.

e) Unity of system:

The entire system should appear as a single working environment so that it will give the system a sense of stability. This will in turn garner trust from users.

f) Reliability:

A measurement of how a disaster could affect the system. A system is considered reliable as long as it does not cause dangerous or irrecoverable damage.

g) Maintainability: -

The system would be developed by means of a modular or top-down approach. A large or complex routine is broken into smaller and manageable modules. This is especially evident in the tutorial section where the knowledge base is separated from the inference engine. This will ensure good maintainability in the future.

h) Understandability: -

Coding in the program should be clear and understandable to other programmers. The program would contain adequate comments as pointers for future programmers for the system.

quickly create complex paging and navigation structures. RTF text can be imported into the framework icon and Authorware will automatically set up a multi-page navigation system. Full support for hypertext also makes for interactive, highly navigable applications. Hypertext can easily link to text, graphics, sounds, digital movies, or other locations in the application. Besides that, this software provides a great deal of modularity, and this is a definite plus for good programming.

All these factors make a Authorware a good software to program an effective learning tool which certainly has to incorporate multimedia features as well as the hypertext capability.

3.3.2 Visual Prolog 5.1

Prolog was originally designed to be an AI language, and it is very well suited for expert systems and similar AI applications. Frame or rule-based systems, forward or backward chaining, pattern-matching systems, and constraint-resolution systems; all are natural and elegant expressions of Prolog's underlying semantics.

Prolog is what is known as a declarative language. This means that given the necessary facts and rules, Prolog will use deductive reasoning to solve your programming problems. This is in contrast to traditional computer languages, such as C, BASIC and Pascal, which are procedural languages. In a procedural language, the programmer must provide step by step instructions that tell the computer exactly how to solve a given problem. In other words, the programmer must know how to solve the problem before the computer can do it. The Prolog programmer, on the other

hand, only needs to supply a description of the problem and the ground rules for solving it. From there, the Prolog system is left to determine how to find a solution.

Because of this declarative (rather than procedural) approach, well-known sources of errors such as loops that carry out one too many or one too few operations are eliminated right from the start. Prolog encourages the programmer to start with a well-structured description of the problem, so that, with practice, Prolog can also be used as both a specification tool, and the implementation vehicle for the specified product.

Yet, the strengths of Prolog extend far beyond the boundaries of AI. The high level of abstraction, and the ease and simplicity with which complex data structures are represented, allow a declarative approach to programming that benefits any discipline or problem solving strategy.

For this reason, Visual Prolog has also been widely used to make administrative applications, advanced database management, planning and scheduling systems and much more.

In Prolog, the number of program lines required to solve a given problem is typically only a fraction of that required by a procedural programming language like C or Pascal. Clearly this is plus point, and since the code is easier to modify, ongoing maintenance would be easier.

	Prolog	C or Pascal
Expert Systems	1	10
Parsers / Compilers	1	8
DB Applications	1	5
Professional UI	1	3
Business Appl.	1	3

Figure 3-1 : Comparison between Prolog and C/Pascal

Furthermore, the proposed system would need to include an intelligent database section. Visual Prolog would be the most suitable software to be used to program this section.

3.4 System Requirements

3.4.1 Hardware

The learning system would be a Windows application that would be able to be operated on typical stand-alone computers. One of the more specific hardware requirements is a CD-ROM/floppy drive. To fully appreciate its multimedia capabilities, a VGA monitor is recommended.

3.4.2 Software

It is important that the package developed does not require any runtime version of additional system software, as this would definitely be a negative factor. Students would be greatly discouraged to use the package since he has to incur additional costs so as not to infringe upon the copyright act. Moreover, schools would have to think twice before embarking on using computer as a teaching aid if too many restrictions are imposed on the usage of the system.

As such, the system should be able to run as a stand alone executable application. All the software it requires is the Windows 95/98 operating system.

4.0 SYSTEM DESIGN

4.1 Hierarchy of the system contents

The design of the system would take into consideration that this is a system that focuses on being an effective learning tool. This learning tool must be able to relate facts, graphics, rules and text to help the understanding of the student.

Generally, the system will consist of 2 main sections:

- Study Section
- Database Section

We will use a Pyramid Analysis Diagram to split up the course contents into manageable chunks. These chunks would then be combined into modules. After that, the order in which to develop them and the media can be decided.

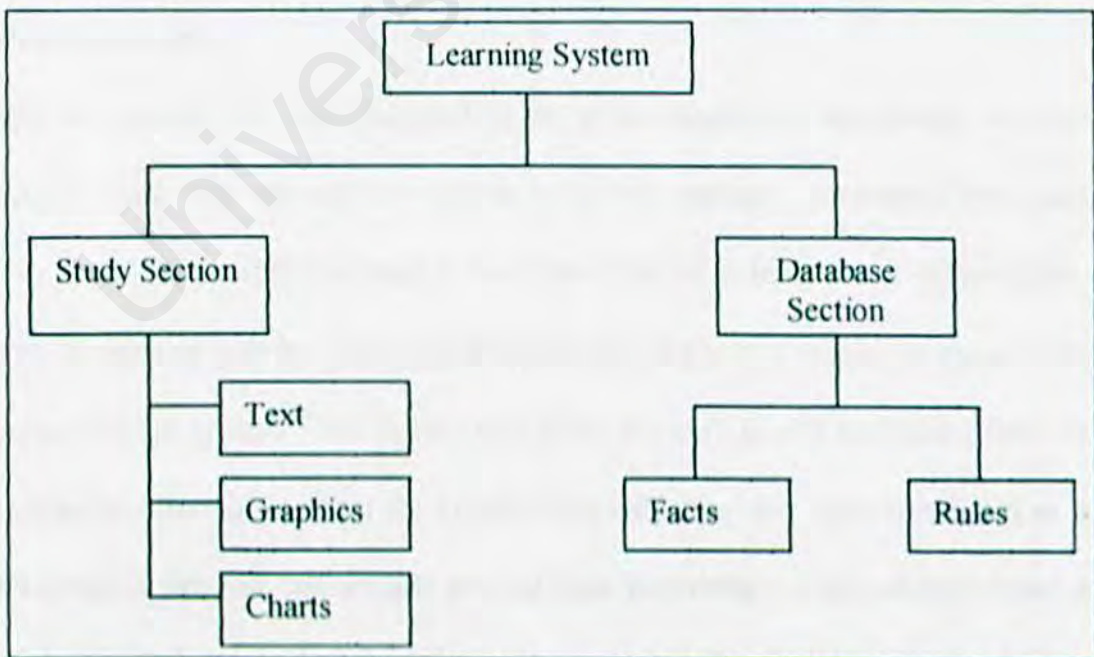


Figure 4-1: Pyramid Analysis Diagram

4.1.1 Study Section

The study section would contain lecturing materials such as notes. The notes will not consist of text alone but will be accompanied with numerous illustrations and graphics. This multimedia approach will help to better explain facts. In fact, some part of the notes can be “spiced up” with interesting facts that relate to our everyday life. Aside from drawing the student’s attention, this relation of facts would surely improve the understanding of the student, as suggested by the *elaboration hypothesis* (Craik & Tulving, 1975).

Besides that, some of the more important terms found in the text would be hyperlinked to their corresponding descriptions and definitions. The use of hypertexts enables the student to see the relationships between terminologies and concepts. This will help build the student’s knowledge and vocabulary required for understanding of the course.

4.1.2 Database Section

The database section will be designed to be a storehouse of knowledge for the pertaining subject. The user will be able to store and retrieve information from this database. It will be designed in such a way that it has a default set of information. However, the system will be liberal in the sense that it allows the user to change the facts present in the system. This feature will allow the user to add additional facts or retract obsolete information from the system. Not only that, this section will act as a medium in which the user can actually test out their knowledge of the subject. Since it will be designed to be both interactive and non-linear, it would hopefully further enrich the learning experience of the user.

4.2 Knowledge Database Design

The design of the knowledge database used in the Database Section has two main components. They are the classification of data and the data set itself. Both of these two have to be there to complement each other.

4.2.1 Classification Module

The knowledge database will be designed initially with a simple classification template. This classification template will then be improved and focused towards the needs of the system. This module will prove to be the more important module between the two as it's design will affect the way data is represented. Think of it as a framework for processing data. A change in the structure of this classification module would mean changing the way data would be represented on the system. As such, it should be designed carefully to not require further changes in the future. This would make the system stable and dependable.

4.2.2 Data Module

With the classification module in place, the system will still remain incomplete without the data to fully represent the information that needs to be shown. Data must be put in according to the structure of the classification module. Any deviation from the given structure, and the data may be rendered totally useless.

The data module will then be fitted with certain features that will enable it to manipulate the database of the system. Besides having a default set of data, the system should be able to introduce additional data as well as being able to remove old and obsolete data. This will ensure a system that is dynamic and versatile.

4.3 Interface Design

Usability in an interface design is very important. If the usability is good, the interface becomes 'invisible' to the end-user and they are left to get on with the task at hand. Contrariwise, bad interfaces stand out as the end-user does battle with them in order to get the tasks done. Hence, the most important design objective is to make the computer system as invisible as possible, so all the user needs to concentrate on are the task he/she wishes to perform.

In order to be usable, an interface needs to do several things:

- Support the end-users tasks
- Provide clear and unambiguous feedback
- Provide information in terms the end-users can understand and interpret
- Be clear and easy to navigate
- Clearly show where the end-user is going
- Allow input to be easily entered

Based on the issues stated above, the proposed system would include interface features such as pull down menus. These menus will be placed near the top of the screen, much like the familiar pull down menus found in most Windows programs. This should be done consistently so the users would not have to look for them. These menus will function as navigators throughout the system, and they will give the system a non-linear navigation, allowing the user to move to any part of the system with ease. Besides that, a text-based input system will also be used to facilitate information retrieval and transfer, mostly in the Database Section.

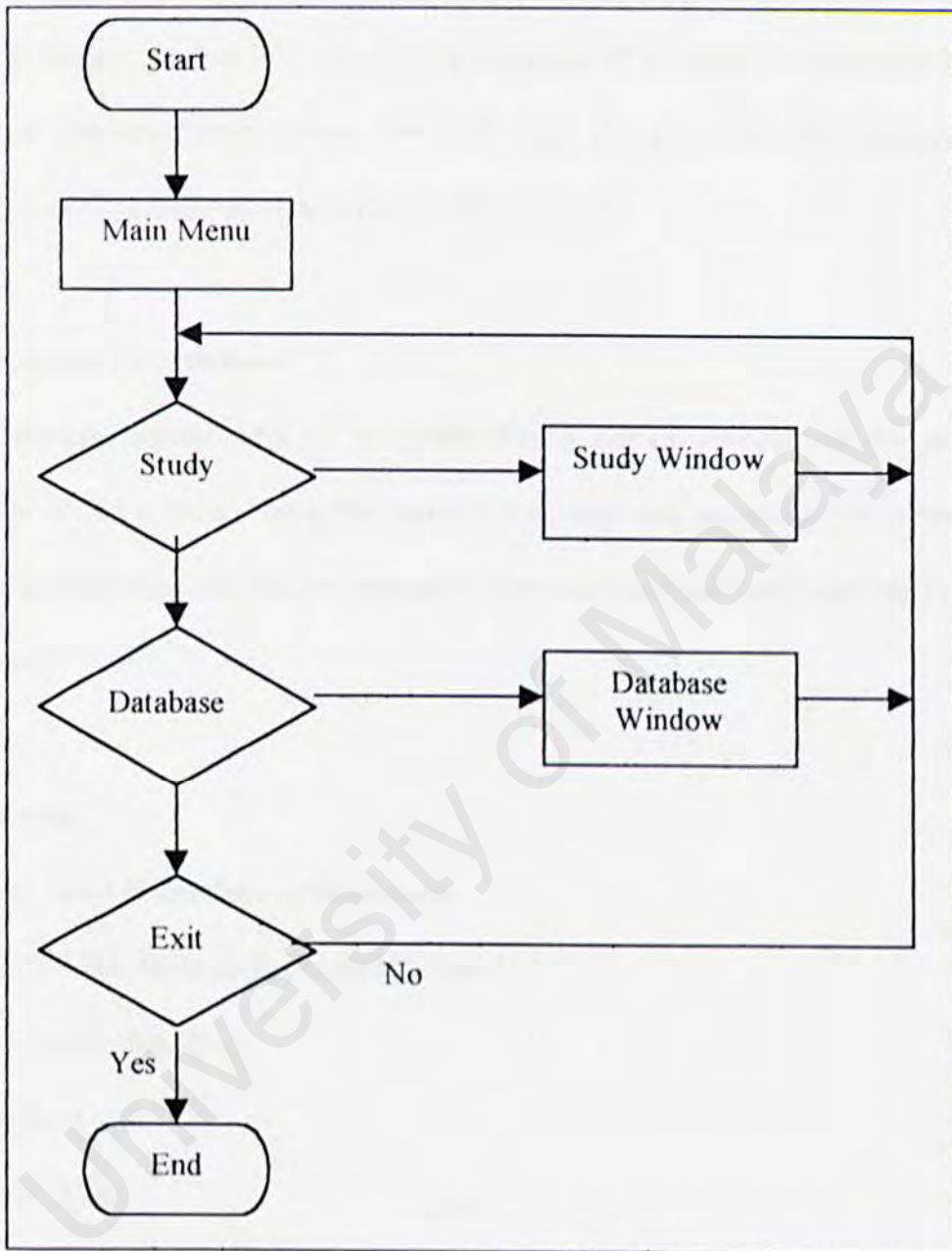


Figure 4-2: A Main Menu Navigation Flowchart

5.0 SYSTEM IMPLEMENTATION AND TESTING

After the design of an entire system, the next logical task would be the coding of the system. The primary goal of this task is the production of a simple, clear source code with internal documentation. Doing this will ease the processes of verification, debugging, testing, modification and further enhancement.

5.1 Development Environment

The development environment of a system has a certain impact on the actual development of the system. Using the suitable hardware and software will speed up the system development and its performance. The hardware and software tools used are as follows.

5.1.1 Hardware

The hardware used in developing the system:

- Pentium II - 266 MHz powered stand-alone PC
- Generic floppy disk drive
- Generic hard disk
- CD-ROM drive
- Standard input/output devices

5.1.2 Software

The software used in developing the system:

- Microsoft® Windows 95
- Microsoft® DOS
- Visual Prolog 5.1
- Microsoft Image Composer

5.1.3 Software Tools Used to Produce Report

The process of documentation is just as important as the actual development of the system itself. Therefore, suitable software must be used to enable proper as well as easy documentation. To this end, a most widely used word processor was chosen:

- Microsoft® Word 97

5.2 Program Development

5.2.1 Reviewing the Program Documentation

Before proceeding with the coding of the system, the program documentation, which was recorded earlier, should be reviewed. The requirements, analysis are all looked at again to reacquaint the developer with the initial objectives and targets for the system. By doing this, any need of possible changes or reinforcements to the requirements and targets can also be detected. The documentation should then be updated with the most recent findings to reflect the additional information and experience gathered thus far.

5.2.2 Designing the Program

At this stage, the basic structure of the program is designed. The design is laid out carefully to help to maintain the focus when implementing the coding later on. Further brainstorming sessions might be necessary to improve the design.

5.2.3 Coding the Program

This is the most exhaustive of the all the development stages as it entails the creation of possibly hundreds of lines of codes, and then reviewing and changing those same lines of codes over and over again. Prototypes of the program are created, tested and reviewed countless times. After that, it will be sent back to the drawing board again to be redesigned and re-coded. When the codes are finalised, the system will be tested again as a whole.

5.2.4 Testing the Program

The program is tested for errors and bugs, and if any are found, noted and will be corrected when returning to the coding phase. Different users are reeled in to help test the system's many aspects.

5.2.5 Document the Program

Finally, the development of the program is documented, complete with the successes of the program as well as the challenges and failures faced.

University of Malaya

5.3 Program Coding

5.3.1 Coding Approach

In this system's coding, it does not involve any procedures (also known as algorithms) found in conventional programs. Instead, it involves rules that use facts to infer new information. This is mainly due to the use of the programming language, in this case Visual Prolog.

The coding style is an important attribute of the source code and it determines the intelligibility of a program. For example, an easy to read source code helps to ease the maintenance and enhancement of the system. Some of the elements of coding style include:

- Internal (source code level) documentation
- Methods for data declaration
- Approach to statement construction

Code documentation begins by deciding that rules in the same category should be placed together. This will ensure the connectivity and neat organization of the program. The use of indentations and empty lines can help to distinguish comments from the actual code.

1. Internal Documentation

Internal comments provide a clear guide during the maintenance phase of the system. Comments provide the developer with a means of communicating with other readers of the source code. Statements of purpose indicating the function of the module and a descriptive comment that is embedded within the body of the source code is needed to describe processing functions. This will greatly help others who want to understand the workings of the system.

2. Naming convention

Naming convention provides easy identification for the programmer. The naming convention is created with coding consistently and standardization in mind.

3. Modularity

In order to reduce complexity and to facilitate change, parallel development of different parts of a system is encouraged. This way, each cluster of code will be independent of each other and can possibly be modified without the need to greatly modify other modules of code.

5.3.2 Coding for the Study Section

The main focus of the Study Section is to disseminate information to the user. As such, an attractive and simple to use Graphics User Interface (GUI) is designed for this purpose. A few important features of Visual Prolog were instrumental in the development of this section.

The Application Expert was used to generate a new project for the system. When a new project is generated, it will automatically set up all the basic tools like a help file, toolbars, menus, etc. The basic source code for this application template was generated automatically.

Also, the VPI (Visual Programming Interface) was selected as the UI (User Interface) strategy for this section. This provides access to a set of Code Experts, resource editors, and additional VPI-specific predicates, definitions, and libraries. The resource editors are used to define, layout, and edit windows, dialogs, menus, tool bars, help bars, string tables, icons, cursors, bitmaps, and run time on-line help. From these, Code Experts generate the necessary Prolog source code framework to present these components in some presentation space (i.e. in a window). This framework is ready to be compiled, linked, and run.

To properly shape the system, windows and dialogs had to be added and used to convey the information that needs to be conveyed. Here is where the Code Experts come in. The code experts of Visual Prolog can help to create standardized code to initiate events, open windows, close windows, create buttons, activate menus and

many other functions. From there, all that was needed was proper implementation of the codes and a little extra coding to complete the Study Section.

The codes for the Study Section are divided into four separate code modules. They are BIOLOGY.PRO, PICTURE.PRO, STUDY.PRO, and VPITOOLS.PRO.

The BIOLOGY.PRO module acts as the main module for the entire system. It is here that the coding for the pull down menus are created. For example, the following codes are used to create the Study Section's main window when the pull down menu labeled "Study" is clicked.

```
task_win_eh(_Win,e_Menu(id_System_study,_ShiftCtlAlt),0):-1,  
    win_main_window_Create(_Win),  
    |.
```

Next, the PICTURE.PRO module is used to store information regarding the pictures that will be displayed. Information such as the size of the picture and the location of where it is to be displayed are kept here. The following code will display a picture with certain specified parameters in a window.

```
win_picture1_eh(_Win,e_Update(_UpdateRct),0):-1,  
    Picture = pict_GetFromRes(idb_picture1),  
    ClnrRCT=win_GetClientRect(_Win),  
    pict_GetSize(Picture, Width, Height, _Size),  
    pict_Draw(_Win,Picture,ClnrRCT,  
rct(0,0,Width,Height),rop_SrcCopy),  
    pict_Destroy(Picture),  
    |.
```

STUDY.PRO is the module used to store all the textual information that is to be displayed. To display the text, a Static Text control is used. An example of the codes is as follows.

```
win_study1_eh(_Win,e_Create(_),0):-!.
win_CreateControl(wc_Text,rct(76,88,313,154),"Textmessage",
,_Win,[wsf_AlignLeft],idct_study1a).
!.
```

The VPITTOOLS.PRO module is automatically generated when invoking the Application Expert. This is a module intended to include the tools selected in the Application Expert.

5.3.3 Coding for the Database Section

The Database Section will be more complex internally when compared to the Study Section as it will involve input, output, knowledge representation, knowledge inferencing and other interactive functions.

This section will be created as a separate project using Visual Prolog's Application Expert. The UI (User Interface) strategy chosen for the creation of this section is the EasyWin UI strategy. It is usually used for small applications that do not need any user interface other than the "current read device", from which it reads input, and the "current write device", to which it sends output.

Because this section is written as a separate application, special code has to be written to integrate it with the other section. The following is the code used to call and run the Database Section from the main pull down menu. This code is found in BIOLOGY.PRO, which is the main module for the entire system.

```
task_win_eh(_Win,e_Menu(id_System_immune_database,_ShiftCtlAlt),0):-  
!  
    W=win_GetParent(_Win),  
    vpi_CreateProcess(W,"immune_s.exe","",wsf_Restored),  
    !.
```

The entire collection of source code for the Database Section is kept in a file named IMMUNE_S.PRO. To accept input from the user, two standard predicates are normally used. The *readln* predicate reads characters from the current read device which is the keyboard as default until it reads an ASCII carriage-return character (ASCII 13). The *readchar* predicate reads a single character from the current read device (which is the keyboard, unless changed via readdevice).

On the other hand, to display input for the user to see, a single predicate is used. The *write* predicate writes the given constants or variables in the current window or on the current write device. For example:

```
write("Item: ",Item),  
nl.
```

To change the facts provided with the system, two other predicates are used. The *assert* predicate inserts a Fact in the matched internal database after any other stored facts for the corresponding predicate. Correspondingly, the *retract* predicate deletes the first Fact in the database that matches the given Fact.

However, to store the information and the changes made to it, a file has to be used for keeping the facts. This text file will be accessed at the beginning of the program to load up the facts into the database. The *consult* predicate reads facts from the text file and asserts them in the unnamed database. And after making changes to the database, the changes need to be saved on that same file too, so that when it is accessed later, the changes can be seen. To do that, the *save* predicate is used. It saves facts from the default database section (declared without a name) into the text file with the specified filename.

5.4 System Testing

After the coding has been completed, the system has to be put to the test. The Study Section is first tested to ensure all the pull down menus are working properly. The items displayed in the windows are tested to make sure that they are displayed correctly.

Then, the Database Section is tested. To test this section, the facts of the original data are first put through the system. Once the results are found to be positive, the system will be further tested with "dummy" data. The purpose of using these "dummy" data is to find out if the system is actually getting the correct results as a product of intelligent reasoning. Even though the system may be correct in those first few cases, there may be some concern that similar successes might not be achieved for other cases, unless the underlying mechanism is proved to be correct.

Besides that, the use of the "dummy" data can also ensure that the system will be able to accept input from the user correctly and make the appropriate changes to the system's database.

6.0 SYSTEM EVALUATION AND CONCLUSION

At the end of the development cycle, this program is evaluated to be potential in fulfilling the expectations, aims and objectives stated earlier. However, due to project constraints and further thoughts, there are still some lingering limitations to the system. This chapter concludes the thesis by describing the evaluation results of this system.

6.1 Problems Encountered and Their Solutions

6.1.1 Difficulty in Choosing a Suitable Programming Language

Initially, when choosing for a suitable programming language to code this system, I encountered a problem. There were just too many programming languages in the market to choose from and all of them looked equally promising.

However, I narrowed them down into a few, based on the information I gathered regarding their common usage. Finally, I settled with Visual Prolog 5.1 because of one main reason - the declarative nature of Prolog. With Prolog, the knowledge base that I planned to create can be properly structured and easily maintained.

6.1.2 Inexperience Using the Programming Language

Although I chose Visual Prolog 5.1 to be used as the programming language of this system, I was not particularly familiar with it. After spending years of study into C++ and Pascal (which are all procedural languages), I could not quite grasp the declarative concept of Prolog. This is compounded with the fact that Visual Prolog 5.1 is actually even more complex than the original Prolog.

However, after reading some more material, I began to tinker with the language. After a certain period of time spent to get to know the language, I was finally able to get a foothold on the basics of this powerful language. Finally, after gaining further experience and with the help of the Code Experts featured on Visual Prolog 5.1, I was able to create a stand-alone application using this language.

6.1.3 Determining the Scope of the System

The field of Biology is immensely wide. To pack everything into this learning system is plainly impossible. Because of that, a proper scope has to be chosen to properly showcase the abilities of this system without focusing on too many areas of the subject. Learning of this subject cannot be too general and must contain some detail. And because detail is required, the scope must be further reduced to match the workload needed to complete this system with the time frame provided.

Therefore, the topic of "Immune System" was chosen since it is a compulsory topic in most Biology lessons especially for the A-Levels. Besides, it is an interesting topic frequently discussed among common people.

6.1.4 My Level of Knowledge of the Subject

Although I have been a student of Biology, I am by no means an expert. So, my understanding of the subject might not be completely accurate. Because of that, the credibility of the system might be affected because there may be incorrect information present in the system.

The solution for this is found in research. To obtain the information required, much research was done regarding the subject. Many books and reputable Internet websites were visited to garner as much information as possible.

6.2 System Strengths

6.2.1 Friendly User Interface

This system will utilise a familiar Windows interface that has been proven to be rather user friendly. The pull down menu and buttons are found at their usual locations. Input is done via the keyboard and wouldn't require too many keystrokes. Basically, the user interface would be quite intuitive and friendly but at the same time not too flashy or distracting.

6.2.2 Ease of Expansion

Because the facts of the Database Section are stored on a separate file, any modifications or expansions to the facts can be done easily and without recompilation of the entire system. In fact, changes to the facts can be done while the system is still running.

Furthermore, this text file which stores the facts can be modified easily by a computer savvy person with a simple text editing software. However, certain rules must be observed when modifying the text file. They are:

- no upper-case characters except in double-quoted strings
- no spaces except in double-quoted strings
- no comments
- no empty lines
- no symbols without double quotes

6.2.3 System Has a Dynamic Database

With the ability to modify and insert new information into the database of this system, the user can actually use the system to store new and updated knowledge regarding the subject provided the structure of the knowledge remains the same. With this feature, the system can actually "grow" and "mature" as the knowledge of the user increases. Conversely, any obsolete information can be flushed out of the system with no trouble at all.

6.2.4 Interactive Environment

As a plus point, the ability to modify the facts initially provided with the system actually gives this system an interactive "feel". Because of this interactive environment, users are more likely to use the system and learn from it, thus creating a better learning environment for students.

6.3 System Limitations and Future Enhancements

6.3.1 Insufficient Information Provided by the System

Mainly due to the constraints of time, the system developed might not have sufficient in-depth information required by most students. This is especially evident in the Study Section where the topics covered are limited and few. Because of that, this system can only be used as a supplementary learning tool and may require further enhancements to be more effective.

6.3.2 Requires Exact Matching

While inferencing for information while using the Database Section, the system would require an exact match (character for character) for a successful solution to be found. Any typo on behalf of the user while inputting will result in a failure. This may be mistaken as a genuine failure by some users and might cause some confusion.

6.3.3 Future Enhancements

The list of enhancements possible for this system is only limited by the perceiving mind. The list of possibilities is endless. This is what makes the system scope so wide in the first place and therefore has to be limited to fit the permitting time frame.

Some enhancements may be introduced to overcome the constraints imposed previously. For example, the number of topics included in the Study Section can be increased to fit in more of the Biology syllabus. Also, the structure of the knowledge base can be improved and further widened to accommodate more immune system components and items.

6.4 Conclusion

The conventional method of studying Biology has several drawbacks. One of the main disadvantage of conventional learning techniques is it frequently causes the student to resort to memorizing facts. This is a poor way of learning and it has to be improved. The proposed system will attempt to change this by incorporating certain intelligent and multimedia features.

From the research done and the literature read, it can be concluded that the use of Artificial Intelligence concepts in the production of Intelligent Computer Aided Learning Systems has great prospects. The academics concerned remain convinced that further ICAL packages would be of immense benefit to the learning process. However, these ICAL packages have to retain certain features such as the multimedia facet, the intelligence of the system and a good user interface. These will all contribute to make learning Biology a fun and interesting adventure.

Besides that, by the use of this software, students will be exposed to the world of computers and Artificial Intelligence. As the world moves toward the information technology era, more and more people are required to be computer literate. With this software, not only will the students learn Biology, they will be sharpening their computational skills as well.

All in all, this system has achieved most of the objectives set forth earlier. However, being a very focused system, it will not be useful if applied to other fields. Although it may not be the perfect computer aided learning environment system, it

will hopefully provide a foundation in which future intelligent systems can be based on.

University of Malaya

REFERENCES

Bernstein, Ruth & Stephen (1989). *Schaum's 3000 Solved Problems in Biology*. McGraw-Hill.

Biology Courseware - <http://www.cyber-ed.com>

Bratko, Ivan (1990). *PROLOG: Programming for Artificial Intelligence*. Addison-Wesley

Computer Aided Learning in Mathematics (CALM) – <http://www.calm.hw.ac.uk>

Coxwell, Joshua, 1996, "Workshop Biology Course",

<http://biology.uoregon.edu/wb.html>

Craik, F. J. M. & Tulving, E. (1975). Depth of processing and the retention of words in episodic memory. *Journal of Experimental Psychology*, 104, 268-294.

Fan, J. P., Mak, T. K. and Shue, L. (1997). A knowledge-based computer instruction system. *Australian Journal of Educational Technology*, 13(2), 98-114.

<http://cleo.murdoch.edu.au/ajet/ajet13/au17p98.html>

ITS - <http://www.es.mdx.ac.uk/staffpages/serengul/table-of-contents.htm>

Jennings, Leonie, 1993, "Research Paradigms in the Social Sciences", Southern Cross University, Lismore.

Jonassen, D. H., 1992, "Cognitive Flexibility Theory and its implications for designing CBI"

Mandl, H. & Lesgold, A. (1988). *Learning Issues for Intelligent Tutoring Systems*. New York.

National Cancer Institute - <http://res.nci.nih.gov>

Report to the President on the Use of Technology to Strengthen K-12 Education in the United States, March 1997

Scrimshaw, P. (1997). Computers and the teacher's role.

Sleeman, D. & Brown, J. S. (1982). Intelligent tutoring systems. New York : Academic Press.

Understanding the Immune System,

http://res.nci.nih.gov/PATIENTS/INFO-TEACHER-immune_sys/Title.html

University of Malaya

1.0 Installation Instructions

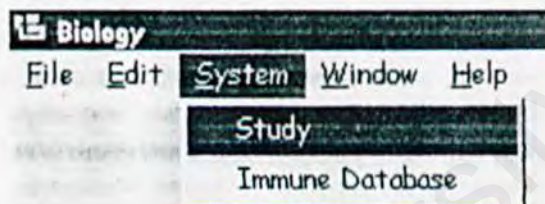
1. Insert the disk labeled "Installation Disk" into your floppy drive.
2. Double-click on the file "Install.exe"
3. Choose a target folder for the files to be installed to and click OK.

When the installation is complete, go to the folder and double-click on the file "Biology.exe" to start the application.

2.0 Operating Instructions

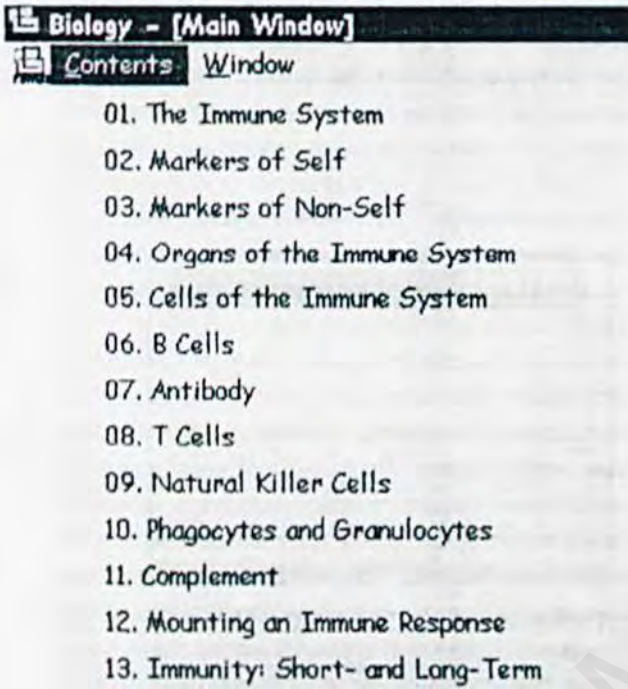
2.1 Study Section

- To go to the Study Section, use the menu: System | Study



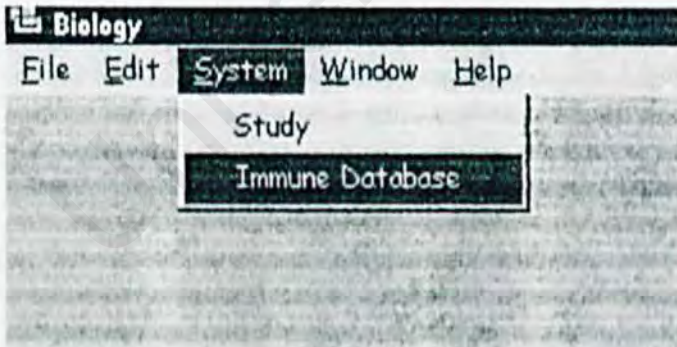
- In the study section, use the menu to navigate to different topics. For **example:**

Contents | 01. The Immune System



2.2 Immune Database Section

- To open the Database Section, use the menu: System | Immune Database



- Once the Database Section window is open, follow the on-screen instructions to properly use the system. When asked for input, please do observe that upper-case characters and lower-characters are treated as different characters.
- When the program ends, please click the control on the top right of the window to close the window.

