INTELLIGENT ASSESSMENT SYSTEM ROBOTIC

NURUL AIN BINTI HASSIM

WEK020194

FACULTY OF COMPUTER SCIENCE
&
INFORMATION TECHNOLOGY
UNIVERSITY MALAYA

SESSION 2004/2005

ABSTRACT

Intelligence is harder to define than knowledge. When researchers in the field of artificial intelligence talk about intelligence in technical systems, they usually use it to suggest that their software is more flexible, more readable, and easier to use than some other software.

Early attempts were limited to only simple grading of assignment, but recent developments have started to focus on adaptive assessment. The assessment systems use adaptive approaches to support intelligent assessment and individualized learning. Various approaches have been used for this purposed and the field is still quite new. Keeping this in my mind, I decided to build the new intelligent assessment system which is adaptive and focusing in robotic domain. This intelligent assessment system is a stand alone system. This assessment system is integrated between three topics in robotic syllabus. There are speech synthesis, motion and vision. The users that are allowed to use this system are either the expert or novice user. The aim of this project is to provide efficient intelligent mediated learning for the student and to provide helps based on the student's categorized either beginner, intermediate or advance student. I believe it's important to build a strong foundation and linking for robotics in order to ensure that will be ready to face new and often exciting challenges that they will encounter in their studies. To love something, we have to like something. And to like something, we have to know something. The Intelligent Assessment System is thus proposed to overcome any shortcomings that other systems in other fields have. The build system will have and attractive user interface to make the

assessment process more efficient. This system will be built using Visual Basic 6 as a programming tool. The methodology that will be using through out this system is waterfall model. And to help lost or inexperienced computer users, the system will be built to cater the needs and help students along the way. In order to build a comprehensive system, related information is gathered from the books, internet and other learning material. It is hoped that the development of this intelligent assessment system will further assist students to better understanding in robotic domain.

TABLE OF CONTENTS

CHAPTER 1-INTRODUCTION

CHAPTER 2-LITERATURE REVIEW

LISTS OF TABLES

LISTS OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Computers have been used in education for over 20 years. Computer-based training (CBT) and Computer aided instruction (CAI) were the first such systems deployed as an attempt to teach using computers. The learner's abilities were not taken into account. While both CBT and CAI may be somewhat effective in helping learners, they do not provide the same kind of individualized attention that a student would receive from a human tutor. For a computer based educational system to provide such attention, it must reason about the domain and the learner. This has prompted research in the field of intelligent assessment systems. Intelligent assessment systems offer considerable flexibility in presentation of material and a greater ability to respond to idiosyncratic student needs. These systems achieve their "intelligence" by representing pedagogical decisions about how to teach as well as information about the learner. This allows for greater versatility by altering the system's interactions with the student. It has been shown to be highly effective at increasing students' performance and motivation.

Thus an effective assessment system based on the resources should be established to promote the teaching and learning process as to improve the performance of the whole education system. In this project, an intelligent assessment system will be constructed. Compared with the current assessment in education, this new intelligent assessment system expands the range of objects for evaluation and takes some AI technologies to give

more heuristic and intelligent assessments. Thus, the Robotic-Intelligent Assessment System is built upon the foundation of digital technology to supplement the traditional teaching.

1.2 INTELLIGENT ASSESSMENT SYSTEM

Assessment is an essential part of learning as it provides a measure of what has already been learnt. Meanwhile the meaning of intelligent itself is it can be referred as the system's ability to know what to teach and how to teach the subject or the course. The intelligent assessment system is able to assess such problems and with the intelligent help to provide the correct answer.

Bork ET. Al provided a comprehensive overview of the assessment practices popular in traditional academic environment. In their opinion, much of the assessment is not done properly and therefore does not contribute to effective learning. Waters and McCracken emphasized that the assessment should not be solely a grade-assignment or ranking tool. The main goal of assessment should be to enhance the learning experience. The traditional assessment tools generally focus on isolated facts and techniques and ignore a student's understanding of the larger integrated picture, allowing success based on rote memorization rather than true understanding and in some cases even encouraging the superficial approaches. Assessment should be used to learn about the gaps in knowledge and mistaken knowledge and it should focus on problem

solving, thinking and reasoning skills. The US National Science Educational Standard proposed that the assessment should be done for authentic tasks which are similar to the tasks performed in "real life".

1.3 PROJECT OVERVIEW

Robotic-Intelligent Assessment System is an adaptive system that able to control the student progress which is, it provides an intelligent help facilities to the student according to their type of user that will be using this system. Basically Intelligent Assessment system is a stand-alone system, which can be run via the any of the stand-alone computer. With this it is hope that the system will provide access to the user that needs help.

In this system, it is allowed participation from the student who will become the user that will use this system. This system provides interactive adaptive communication between the system and the students.

In conclusion, this system can achieve both it is intellectual objectives as well as it is technological objectives. The benefits can be quite different to face-to-face tutoring. Therefore for the generation of today and tomorrow, this Intelligent Assessment System is the appropriate form of the technology for managing and supporting student's learning in robotic domain which it is a new area in education.

1.4 PROJECT MOTIVATION

The main motivation for developing this project is the inefficient communication among students and the lecturers. There are a lot of problems aroused among the students when they need somebody or something to ask and share their problems on studies. There are three types of helper that a student can use nowadays. The channels are the teachers, friends and books. But for both teachers and friends, they are insufficient helper people who have a limitation to give solutions immediately on time. This is because human expertise may become unavailable during the time we need them. A student has difficulty to meet their lecturer, who is busy with works, meeting or lecturer located at other country to get some additional clarification or explanation. So student may have trouble in making a solution of study's problem or to make a decision.

At the same time many of the students liked to refer a book rather than to meet a lecturer. They might think it is hard to see a lecturer when they need immediately help so the fastest way is to use book. When the students wanted to find an answer for the question given, they do not find it in a single book. They must use another book to get the answer so it would cost the student more to find solution in another book. Other problems occurred when there is a solution, but they do not know which the precise answers are and also there is not enough explanation for them to understand. Another limitation when the student used book it is might

be limited examples to show for the student. With the system project that might be develop soon will help the student on their studies hopefully.

1.5 PROJECT OBJECTIVE

The aim of this project is to provide an efficient intelligent mediated learning for the student. This adaptive system also provides feasible, friendly and conducive environment for the student to discuss their assessment through the help thus establishing a suitable environment for teaching and learning process. This is able to provide a convenience, accessibility and quality interaction for the students. It also provides an efficient learning environment to the students who may or may not have prior exposure to the subject contents.

Meanwhile the objectives for this project is more to individual guidance according to student skills with add an intelligent itself to the software tools to provide help or a degree of support to students, that enabling them to work by themselves. There are three types of helps that will be using in this project based on the three types of students which can be classified as the beginner, intermediate and advance students. For an example, if the student login into the system, the student must seat for a pretest test to identify which type of classes they have been classified. Then, based on their result when taking the pretest, they are classified again to take which topic they want by measure their result and divided it into three categories which is low, medium and high result. If the voice communication pretest

result is low, the first help is enabling or if vision in a medium result, the second helps enable and so on. From that, student can learn more effectively when their taking notes and tutorial.

The purpose is to make sure all the help provided are well used of the students and to investigate the effectiveness of an intelligent system in providing help. The other objectives are to use Intelligent Hypermedia with the knowledge base that will build and to rectify student misconception.

Another objective to build this project or Robotic – Intelligent Assessment System is to promote self study among the student using this assessment system.

1.6 PROJECT SCOPE

Generally the becoming assessment system that will develop is focusing in robotic domain. It's represent the knowledge domain in robotic that the student learns in class. The three topic in robotic will be integrated. There are speech synthesis, motion and vision.

Intelligent Assessment System is expert-based systems which must be able to recognize errors and misconceptions, to monitor and intervene when necessary at different levels of explanation, and to generate problems on a given set of instructional guidelines.

It's also including an idea about fundamental of robotic such as description, explanation, problem solving, path planning, simulation and

geometric transformation for the robotic domain. It's allowed the student to give an answer during the pretest to determine which type of student will be classified and the system will diagnosis either the student is beginner, intermediate and advance student based on their marked when seat for pretest. During the student take their pretest to classified type of student, student are required to answer a multiple types of questions such as fill in the blank, drag and drop, true false and objective. During answering true false and objective questions student will given another form that are based on fuzzy logic style that required the student to select their level of confidence when answering the question.

The student next can proceed through their type of student either to take an assessment or read notes that provided. The system also rectifies student misconception with guide them through the intelligent help provided while taking an assessment. The intelligent helps provided are based on their types of student.

This system contains four modules.

- Student module
- The User Interface module
- Learning module
- Assessment module

1.7 AI TECHNIQUES

- Cognitive Science
- Robotic
- Geometric Transformation
- Student Modeling
- Problem Solving
- Simulation
- Fuzzy Logic

1.8 PROJECT LIMITATION

While Intelligent Assessment System somewhat successful on a small system, several problem must be overcome before they have widespread impact. Several authors (eg; Psotka, Massey and Mutter, 1998) have discussed a wide range of limitation on Intelligent Assessment System. Most Intelligent Assessment System has focused on subjects taught in typical primary and secondary subjects. In this context, probably the most significant limitation of Intelligent Assessment System is they can be developed only for a few topic areas (Wenger, 1987). Effective Intelligent Assessment System requires virtual omniscience, which means the compute mastery of the subject area the Intelligent Assessment System are to tutor and covered, including an understanding of likely student misconception. Thus, the most successful Intelligent Assessment System

has been developed for simple procedural skills like solving short problem in mathematic, science and logic.

One of the main difficulties in designing Intelligent Assessment System is the time and cost required. A large team including computer programmers, domain experts and educational theorist, is needed to create just one Intelligent Assessment System.

1.9 PROJECT SCHEDULE

A project schedule gives rough estimation of the frame that will be needed to complete all the individual tasks. The schedule can help to determine whether the project can be finished in time. It will distribute the tasks within a fine time to enable the best use of resources.

This project schedule is planned as a guideline to manage the various tasks that has to be completed within the given duration or risk undertaking unnecessary additional cost of main power.

Figure 1.1Gantt chart explained project runtime

1.10 REPORT OVERVIEW

Chapter 1 – Introduction

This opening chapter describes an overview of the project's proposal. This includes briefly explanation about intelligent assessment system, objectives, overview, scope and limitation of the project.

Chapter 2 – Literature Review

This chapter includes the literature review of other works on associated issues and field of interest. The main topic consists of the study on the difference in the similar type of the assessment system. It also describes about further reading in considering what the tools that will be used to develop this project.

Chapter 3 – Methodology

This chapter identifies and provides the clarification on the methodology, mechanism and approach that will be used in this project.

Chapter 4 – System Analysis

This chapter describes detailed about functional and non functional requirements that are needed in this project. A detailed analysis of the system development tools is covered which include the practicality and effectiveness of the chosen tools use in this project.

Chapter 5 – System Design

This chapter includes the fundamental design such as the data flow approach, functional design and also the user interface.

Chapter 6 – System Implementation

This chapter discuss about the steps and methods taken in implementing earlier proposed design unto developing the system. The developed system will be tested for functionality and errors.

Chapter 7 – System Testing

This chapter describes detail about the testing phase in the development of the system.

Chapter 8 – System Evaluation and Conclusion

System evaluation and conclusion discusses about the problem faced during the system development and also the significance of the system.

References

All the references when built this system.

User Manual

The user manual provides to the users to help them to navigate through the system. Every major function of will be described in this manual.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In the process of developing Robotic-Intelligent Assessment System, the part of research is very important. Research has been done to understand various new concepts, which especially focus on the information. The purpose of the literature review is to get better understanding on what kind of mandatory requirements for the system in order to determine what features that can be added to the system. Besides that, better understanding about the development tools for developing the system will be gained and will get better knowledge on the development methodologies used while developing the system. The research on the current online assessment system also has been carried out. All these information gathering was done by using a number of methods such as surf the Internet, reading from newspaper, books and other relevant materials.

2.2 APPROACHES TO LITERATURE REVIEW

A few ways have been identified to do research in completing this intelligent assessment system as shown below:

- References

  Examples of references are material such as books, magazines, journals, newspapers and previous senior's thesis in the document room. All these references were read through for new ideas about the assessment system. Technologies were analyzed to see if they are suitable in the current system's environment. These sources

offer much information regarding the latest technologies currently in the market. Nevertheless, the information is useful for future development of the system.

- Internet Research

  Internet is used as the main resource for referring any ambiguities that arise during the entire development period. By analyzing the similar assessment system through research on the World Wide Web has help in giving ideas on the features, functionality as well as the design of the web-based system.

2.3 CURRENT TRENDS IN THE APPLICATION OF AI IN EDUCATION

In the last several years, applications of AI in education have diversified, approaches are more fractured, and the field is certainly not as unified as five years ago when Intelligent Tutoring System (ITS) dominated AI research in education. These new trends exemplify different responses to the difficulties encountered in developing and testing the first generation of ITS. Very broadly, while some recent research attempts to improve the one-on-one tutoring method of teaching associated with ITS, other work is investigating different methods of teaching and learning. This research is rethinking important principles of learning and teaching that should underpin education in general and the design of educational technology in particular. At the same time, new work is attempting to expand the range

of learning goals and outcomes associated with AI-based systems for education. In some cases this means developing educational software for a more diverse set of subjects, but in other cases the targeted knowledge has less to do with the subjects learned than with the quality or depth of learning.

A substantial amount of work continues within the ITS framework. Different groups are attempting to improve the various components of ITS, to develop applications in increasingly complex subjects, and to make ITS more cost-effective to develop by providing "shells" and other system-building tools that institutionalize the basic structure of ITS. In doing so, this research largely holds fixed the drill-and-practice method of teaching, while attempting to enhance learning outcomes or goals either by improving the quality of tutoring of subjects already within range of ITS, or by expanding the range of subjects ITS can tutor.

While many early ITS focused mainly on simple topics in high school mathematics, recently ITS have been developed for more advanced topics in mathematics (Du and McCalla, 1991) and science (Lester and Porter, 1991). ITS have also grown beyond mathematics and other more formal subjects to include topics in history, language and social science. Bruneau, Chambreuil, Chambreuil, Chanier, Dulin, Lotin and Nehemie (1991) describe the design of a tutor for reading; Frederiksen, Donin, DeCary and Edmond (1991) are developing a tutor for second language learning; and Feifer (1989) has developed a tutor that not only helps students learn to

read but also focuses on inference and knowledge-structuring strategies. Similarly, ITS have diversified beyond public school curricula to topics in training and vocational education. For example, new ITS for electronics, maintenance, and troubleshooting (Cooper, 1991; Frederiksen, White, Collins and Eggan, 1988; and Kurland and Tenny, 1988) have built on the seminal work on SOPHIE (Brown, Burton and deKleer, 1982).

My job in this project is to create new domain area which is robotic that will contribute in the learning education to the student

2.4 9 Principles of Good Practice for Assessing Student Learning.

1) **The assessment of student learning begins with educational values.**

Assessment is not an end in itself but a vehicle for educational improvement. Its effective practice, then, begins with and enacts a vision of the kinds of learning we most value for students and strive to help them achieve. Educational values should drive not only *what* we choose to assess but also *how* we do so. Where questions about educational mission and values are skipped over, assessment threatens to be an exercise in measuring what's easy, rather than a process of improving what we really care about.

2) **Assessment is most effective when it reflects an understanding of learning as multidimensional, integrated, and revealed in performance over time.**

Learning is a complex process. It entails not only what students know but what they can do with what they know, it involves not only knowledge and abilities but values, attitudes, and habits of mind that affect both academic success and performance beyond the classroom. Assessment should reflect these understandings by employing a diverse array of methods, including those that call for actual performance, using them over time so as to reveal change, growth, and increasing degrees of integration. Such an approach aims for a more complete and accurate picture of learning, and therefore firmer bases for improving our students' educational experience.

3) **Assessment works best when the programs it seeks to improve have clear, explicitly stated purposes.**

Assessment is a goal-oriented process. It entails comparing educational performance with educational purposes and expectations; those derived from the institution's mission, from faculty intentions in program and course design, and from knowledge of students' own goals. Where program purposes lack specificity or agreement, assessment as a process pushes a campus

toward clarity about where to aim and what standards to apply, assessment also prompts attention to where and how program goals will be taught and learned. Clear, shared, implementable goals are the cornerstone for assessment that is focused and useful.

4) **Assessment requires attention to outcomes but also and equally to the experiences that lead to those outcomes.**

Information about outcomes is of high importance; where students "end up" matters greatly. But to improve outcomes, we need to know about student experience along the way, about the curricula, teaching, and kind of student effort that lead to particular outcomes. Assessment can help us understand which students learn best under what conditions; with such knowledge comes the capacity to improve the whole of their learning.

5) **Assessment works best when it is ongoing not episodic.** Assessment is a process whose power is cumulative. Though isolated, "one-shot" assessment can be better than none, improvement is best fostered when assessment entails a linked series of activities undertaken over time. This may mean tracking the process of individual students, or of cohorts of students, it may mean collecting the same examples of student performance or using the same instrument semester after semester. The point is to monitor progress toward intended goals in a spirit of continuous

improvement. Along the way, the assessment process itself should be evaluated and refined in light of emerging insights.

6) **Assessment fosters wider improvement when representatives from across the educational community are involved.**

Student learning is a campus-wide responsibility, and assessment is a way of enacting that responsibility. Thus, while assessment efforts may start small, the aim over time is to involve people from across the educational community. Faculty plays an especially important role, but assessment's questions can't be fully addressed without participation by student-affairs educators, librarians, administrators, and students. Assessment may also involve individuals from beyond the campus whose experience can enrich the sense of appropriate aims and standards for learning. Thus understood, assessment is not a task for small groups of experts but a collaborative activity; its aim is wider, better-informed attention to student learning by all parties with a stake in its improvement.

7) **Assessment makes a difference when it begins with issues of use and illuminates questions that people really care about.** Assessment recognizes the value of information in the process of improvement. But to be useful, information must be connected to issues or questions that people really care about. This implies assessment approaches that produce evidence that relevant parties

will find credible, suggestive, and applicable to decisions that need to be made. It means thinking in advance about how the information will be used, and by whom. The point of assessment is not to gather data and return "results"; it is a process that starts with the questions of decision-makers, that involves them in the gathering and interpreting of data, and that informs and helps guide continuous improvement.

8) **Assessment is most likely to lead to improvement when it is part of a larger set of conditions that promote change.** Assessment alone changes little. Its greatest contribution comes on campuses where the quality of teaching and learning is visibly valued and worked at. On such campuses, the push to improve educational performance is a visible and primary goal of leadership; improving the quality of undergraduate education is central to the institution's planning, budgeting, and personnel decisions. On such campuses, information about learning outcomes is seen as an integral part of decision making, and avidly sought.

9) **Through assessment, educators meet responsibilities to students and to the public.**

There is a compelling public stake in education. As educators, we have a responsibility to the publics that support or depend on us to provide information about the ways in which our students meet

goals and expectations. But that responsibility goes beyond the reporting of such information, our deeper obligation, to ourselves, our students, and society is to improve. Those to whom educators are accountable have a corresponding obligation to support such attempts at improvement.

2.5 INTELLIGENT ASSESSMENT SYSTEM

The rationale of computer-based teaching systems is that, for given students and topics, a computer system can alleviate the variance of human-based teaching skills and can determine the best manner in which to present individually targeted instruction in a constrained subject domain. In order to minimize the discrepancy between a user's knowledge state and the representation of an identified expert's knowledge (a 'goal state') the IAS must be able to distinguish between domain-specific expertise and tutorial strategy.

2.6 Why should we have adaptive Intelligent Assessment systems?

The concept that computer systems should be capable of adapting themselves to suit the needs either of individuals or of different classes of users is an apparently attractive, if slightly unusual, one. Computer systems can be difficult to learn and once learnt, may be easily forgotten.

As the proliferation of computer applications and different delivery platforms continues apace, the number of individuals exposed to the vagaries of a range of computer environments likewise continues to grow. Computer applications tend to embody specific characteristics which make the chose design solution better suited to some users than others. However, many systems could potentially have a range of different features to match the diversity of user populations.

Adaptive systems are systems which can alter aspects of their structure, functionality or interface in order to accommodate the differing needs of individuals or groups of users and the changing needs of users over time (Benyon, Innocent and Murray, 1987). Adaptive systems seek to take over the burden of tailoring systems to individuals and groups.

2.7 STUDENT MODELING

Student modeling is one of the most important topics of ITS research, because the behavior of an ITS largely depends on a student model which represents a snapshot of the student's knowledge. It formulate the student modeling problem as an inductive inference problem, i.e., a problem of constructing a model explaining observed data which are, student's answers to the problems given.

Tutoring is to guide students toward a better understanding of teaching material. This means that the learning process is essentially attained

through the change of their minds and hence the consistency of student's answers can be easily lost. Therefore, student modeling methods should be able to automatically manage the consistency of student's answers in order to follow the student's change. Contradictions which a modeling system should cope with are classified into the following two types:

(1) Contradictions which should be resolved by revising the student model, and

(2) Contradictions which should be captured as they are.

A student modeling system should realize flexible modeling behavior and construct reasonable student models from didactic viewpoints by embodying a teacher's insight, e.g., the ability to capture her student's status by asking fewer questions. In order to cope with contradictions of type (1) which inevitably appear in the student modeling process, a student modeling system is required to have the ability to cope by making belief revisions to keep data for inference consistent.

2.8 Comparative Study of ITSs and Traditional CBI Systems

Seidel and Park (1994) described several intelligent features which are intrinsic to ITS, that are difficult or impossible to include in traditional CBI using ordinary programming techniques.

- ITS can generate knowledge, rather than selecting pre-programmed frames containing knowledge, to present to the student spontaneously according to the student's on-going needs during the instructional process.
- ITS allows both the system and student to initiate instructional activities by applying AI techniques. This "mixed initiative" approach is an important intelligent feature to simulate the live one-to-one tutoring process.
- ITS can make inferences in interpreting the student's inputs, diagnosing misconceptions and learning needs, and generating instructional presentations on the basis of what is available that time.
- ITS can monitor, evaluate and improve its own performance by applying AI techniques commonly used in machine learning.
- Modelling of the student learning process and qualitative decision-making of instruction are intelligent features of ITS.

A comparison between ITS and traditional CBI systems (Seidel & Park, 1994) is presented in table 2.1.

ISSUE	CBI	ITS
Development Goal	Practical use for instructional improvement	Exploration of new technology (AI) in instruction
Theoretical Base	Existing theories and principles of learning and instruction	Information processing theories and cognitive science
System Structure and Process	Frame-oriented static and single structure; prespecified algorithmic system-initiative process	Process-oriented dynamic and modular structure; generative, mixed initiative process
Instructional Principles	Various, but mostly program-initiated expository approach	Mostly student-centered discovery approach
Knowledge Structuring Methods	Mainly task analysis for identifying subtasks and content elements	Knowledge representation techniques for structuring knowledge into the system
Student Modelling Method	Binary judgement or quantitative assessment	Reasoning and qualitative evaluation

ISSUE	CBI	ITS
Instructional Formats	Mostly expository tutorials drill and practice, games and simulations	Mostly inquiry tutorials, games, and simulations
Subject Matter Areas	Virtually every area	Limited to well-structured areas
Development Process and Team	Systems approaches; instructional designer, subject matter expert, and programmer	Prototyping approach; Mostly AI expert (or knowledge engineer) only
Evaluation	Formative and summative evaluation; instructional effectiveness	Validation of functional running; mainly technical debugging

Table 2.1 Comparison between Traditional and Intelligent Assessment

2.9 Domain of Robotic

During literature review, I had done further reading about the domain that I will develop in the Intelligent Assessment which my domain is robotic. Even though my assessment is integrated between the three topics of robotic which are motion, speech synthesis and vision but here I will let you to know about the definition of a robot, laws of robot and benefit that we can gain from a robot.

Definition of a "Robot"

A dictionary defines "robot" as a mechanical device that sometimes resembles a human, and is capable of performing a variety of often-complex human tasks on command, or by being programmed in advance. According to the Robot Institute of America (1979) a robot is: "A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks".

A more inspiring definition can be found in Webster. According to Webster a robot is:

"An automatic device that performs functions normally ascribed to humans or a machine in the form of a human."

Three laws of robotic

Law Zero:

A robot may not injure humanity, or, through inaction, allow humanity or through inaction, allow community to come to harm.

Law One:

A robot may not injure a human being, or, through inaction, allow a human being to come to harm, unless this would violate a higher order law

Law Two:

A robot must obey orders given it by human beings, except where such orders would conflict with a higher order law.

Law Three:

A robot must protect its own existence as long as such protection does not conflict with a higher order law.

Benefits from Robot

Robots offer specific benefits to workers, industries and countries. If introduced correctly, industrial robots can improve the quality of life by freeing workers from dirty, boring, dangerous and heavy labor. it is true that robots can cause unemployment by replacing human workers but robots also create jobs: robot technicians, salesmen, engineers, programmers and supervisors. The benefits of robots to industry include improved management control and productivity and consistently high quality products. Industrial robots can work tirelessly night and day on an

assembly line without a loss in performance. Consequently, they can greatly reduce the costs of manufactured goods. As a result of these industrial benefits, countries that effectively use robots in their industries will have an economic advantage on world market.

2.10 FINDINGS (Existing System)

2.10.1 Development of ITSs in Context of CAL

In the 1950's, Skinner (1958) proposed simple 'Linear Programs', based on the principle of operant conditioning. Material in these programs was presented step by step to the student in a series of frames. Most frames had simple questions with immediate feedback. Programs proceeded regardless of students' understanding of previous frames. These programs did not provide individualization; all students received exactly the same material in exactly the same sequence irrespective of their abilities, background, and previous knowledge of the domain. Carbonell (1970) commented that with this type of systems '*the computer does little more than what a programmed text book can do*'.

Crowder (1959) proposed using student's responses to control the material shown to the student. The 'branching programs' that resulted still had a fixed number of frames, but were able to comment on a student's response and then use it to choose the next frame. The main features of these programs were that they offered corrective feedback, and adapted the selection of teaching material to the students.

In the late 1960's and early 1970's 'generative systems' or so-called 'adaptive systems' came into picture in which meaningful problems could

be generated and solved by the computer. The intention was to do away with all the pre-stored teaching material, problems, solutions and associated diagnostics, and actually generate them. This drastically reduced the memory usage and systems could generate and provide as many problems as the student needed to some desired level of difficulty. Uhr (1969) implemented a series of systems which generated problems in arithmetic that were 'tailor-made' to a student's performance. Suppes (1967) and Woods & Hartley (1971) produced systems with similar abilities. These programs were restricted to drill-type exercises in the domain as well as structure. They did not possess any real knowledge of the domain and they could not answer questions. The gap between the student's cognitive processes and the internal workings of the programs was too wide (Sokolnicki, 1991). Only parametric summaries of behavior were used to guide problem generation, rather than an explicit representation of the student's knowledge (Sleeman & Brown, 1982b). The model of the student was primitive, sometimes consisting of only an integer to indicate the level of the student's competence. Yazdani (1986) notes that

'None of these systems (CAI) has human-like knowledge of the domain it is teaching, nor can it answer the serious questions from the students as to "why" and "how" the task is performed.'

Hawkes et. al. (1986) noted that CAI systems were lacking of many reasons.

- ❖ They attempted to produce total courses rather than concentrating on building systems for more limited topics.
- ❖ They had severe natural language barriers which restricted user interaction with them.
- ❖ They had no 'knowledge' or 'understanding' of the subject they tutored or of the students themselves.
- ❖ They tended to be static rather than dynamic. There was little experimentation with systems in order to improve them.

Self (1974) argued that a computer tutorial program should have a representation of what is being taught, who is being taught and how to teach the student. ITSs should dynamically analyze the solution history and use principles to decide what to do next, rather than requiring solutions to be anticipated by the author of the program. Providing a truly 'intelligent' system was recognized to be a non-trivial task which needed experts from several disciplines.

- ❖ ITSs provide a clear articulation of knowledge for a limited domain.
- ❖ ITSs have a model of student performance which is dynamically maintained and is used to drive instruction.
- ❖ The ITS designer defines the knowledge and the inferable rules, but not the teaching sequence, which is derived by the program.

- ITSs provide detailed diagnostics of errors rather than simply drill and practice.
- Students can pose questions to an ITS.

ITS	Developer	Year	Domain Key	ITS ISSUE
SCHOLAR	Carbonell	1970	Geography	1st ITS, N. L. Dialogue
WHY	Stevens, Collins	1977	Meteorology Socratic	Dialogue, Tutoring
SOPHIE NLP	Brown, Burton	1977	Electronics	Interface, Black-Box
WUSOR	Goldstein	1979	Game Strategy	Overlay Structure
GUIDON	Clancey	1981	Mycin Tutor	Expert Systems, Glass-Box
WEST	Burton	1981	Game Strategy	Coaching, Example-based
BUGGY	Brown	1981	Arithmetic	Represent incorrect knowledge
DEBUGGY	Burton, VanLehn	1982	Arithmetic	Off-line diagnostics
STEAMER	Stevens, Hollan	1983	Navy	boiler design Simulation, Mental models
LMS	Sleeman	1984	Algebra	MAL-Rules
MENO	Woolf	1984	Meteorology.	Programming Discourse management
PROUST	Johnson	1984	Programming	Intention diagnosis
ACTP	Anderson	1984	Lisp tutor	Cognitive modelling
SIERRA	VanLehn	1987	Arithmetic	Bug prediction
SHERLOCK	Lesgold, Katz	1991	AF Electronics	Cognitive apprenticeship

Table 2.2 provides some of the historically prominent ITS efforts (Kaplan & Rock, 1995).

2.10.2Another Existing System

Below, there are few another existing systems that used AI techniques to develop Intelligent Tutoring System.

2.10.2.1 LISP TUTOR

LISP TUTOR (Anderson and Reiser 1985) is an Intelligent Tutoring System developed to teach the basic principles of programming in LISP.

In the LISP TUTOR the expert model was created as a series of correct production rules for creating LISP programs and a learner model was built as a subset of these correct production rules along with common incorrect production rules (Holt *et al* 1991).

LISP TUTOR is based on the principle of "learning by doing" where the learner discovers the productions while working through problems. The tutor acts as a problem solving guide but never states the productions to be learned.

LISP TUTOR is an application of Andersons ACT* theory (Anderson 1983). ACT* theory is one of the earliest attempts to establish a complete theory of human cognition. It combines declarative knowledge in the form of semantic nets with procedural knowledge in the form of production rules. In ACT*

learning is accomplished by forming new procedures through the combination of existing production rules.

The main principles of the ACT theory are:

- Cognitive functions can be represented as a set of production rules. The use of a production depends on the state of the system and the current goals.

- Knowledge is learned declaratively through instructions. The learner must carry out the process of knowledge compilation if the productions are to be properly understood and integrated into their existing knowledge and later recalled and used.

Anderson and his team used GRAPES (Goal Restricted Production System Architecture) to represent the knowledge in LISP TUTOR as approximately 325 production rules. The system also embodies around 425 buggy production rules which represent misconceptions which any novice programmer can easily have.

LISP TUTOR employs model tracing to provide a learner with detailed feedback. The learner is given a problem and the tutor monitors the learners input character by character. The tutor generates all the possible next characters using both correct and buggy production rules.

- If the character is predicted by the correct rule the learner is allowed to continue.

- If the character is predicted by a buggy production rule remedial instructions is given.

- If the character is not predicted the tutor says that it cannot understand and asks the learner to try again. After several tries the tutor explains the next step.

This method has the advantages of early diagnosis of learner misconceptions and of giving immediate feedback to the learner. The learner never strays far from a correct solution. However, this can be viewed as unnecessarily restrictive and counter productive as the student is never allowed to explore incorrect behavior.

2.10.2.2 ANDES TUTOR

Andes is an intelligent tutoring system in classical physics that is being developed by researchers at the Learning Research and Development Center (LRDC) at the University of Pittsburgh and the United States Naval Academy (USNA). Andes allows students to solve physics problems in an environment that provides visualization, immediate feedback, procedural help, and conceptual help. The assessment was done in three parts:

1) Free response examination questions

Free response examination questions were the methods for assessing the effectiveness of Andes as a problem-solving tutor.

2) A portfolio of all of the work done by the students on Andes

A second form of assessment which can be applied to Andes is a portfolio method using the students' log files. Every keystroke a student makes, whether it is correct or incorrect, is recorded in a log file. At the end of a session, when the student exits from Andes, this file is uploaded to a local server and saved.

3) A written survey of the participants in the experiment.

Student reaction to the use of Andes in fall 2000 was recorded by use of a questionnaire and from comments made on end-of-course evaluation forms. The questionnaire data were examined as one set for all students in the Andes group.

2.10.2.3 The Aplusix System

APLUSIX (Bouhineau et al., 2002) is a learning environment for formal algebra. This system includes an advanced editor of algebraic expressions that displays the expressions in the usual form and allows the modification of the expressions in this form. This editor is based on the structure of algebraic expressions

(Kieran, 1991) for the upper functions (selection, cut, copy, paste, drag and drop).

Aplusix is an example driven or reasoning driven. For the first step, the problem solution is given by the system. The student needs to give an explanation about the fact or knowledge. Meanwhile for the second step, problem solution is done by the student. Student needs to choose transformation that possible for the problem then the system will display the calculation.

The aim of Aplusix is to provide help for the student on how to learn comparison operation to develop an efficient strategy. Its help in reducing stress while the student try to solve the equation by help of the system.

2.10.2.4 ISIS TUTOR

ISIS-Tutor is an intelligent learning environment to support learning the print formatting language of the well known information retrieval system CDS/ISIS/M for IBM PCs (ISIS for short). This system is supplied by UNESCO and used widely in Russia and in many information centres in the world. The print formatting language is key to many CDS/ISIS operations and mastering the language is important for effective use of the system. In some ways, it is a kind of programming language. To display or print the result of a search, or the content of a database, an ISIS

user has to write a sequence of print formatting commands, really, a more-or-less complex program in print formatting language. This format program is used by ISIS produce an external presentation of the record when displaying or printing it. To print the selected records of a database in the specified format ISIS applies the print formatting program to every record being printed. Print formatting commands, for example, can type a field of the current record or a part of field, can manipulate the current output position, type a constant character string, and so on. Print formats are also used in indexing and sorting. There are over 50 different commands and modifiers in the print formatting language, so a tutoring system for the language is really helpful.

2.11 FUTHER READING

Considering on Operating System and Programming Language

1. Operating System

   - Windows XP Professional

     This operating system was developed by Microsoft and remains the most popular choice to date. With the Windows XP Professional version the file system has been changed from FAT32 to NTFS, allowing a more secure framework for

business online. Its added security features and overall case handling.

- UNIX

  UNIX is a computer operating system. The basic software running on a computer, things like word processors and spreadsheets. It is the most common operating system for servers on the Internet.

- LINUX

  Linux is an operating system that was initially created as a hobby by a young student, Linus Torvalds, at the University of Helsinki in Finland. Linux and open source software offers cost savings, flexibility and better security, according to ITV's director of operations and infrastructure, but is not always appropriate and needs to be assessed on a case-by-case basis.
  Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world. The source code for Linux is freely available to everyone.

2. Programming Language

- Visual Basic

  The Visual Basic programming language is based on BASIC (Beginners All-Purpose Symbolic Instruction Code), a language design by John G. Kemeny and Thomas Kurtz professor of Dartmouth College in early 1960s.

  BASIC is so popular mainly because all of its commands are easy to understand and follow. The language itself is similar to English unlike language such as C, which is arcane at most time can read a program in Basic like a book.

  Visual Basic is a compiled language totally different from its predecessor – the BASIC language, which is an interpreted language.

- Prolog

  Prolog is a logical and a declarative programming language. The name itself, Prolog, is short for PROgramming in LOGic. Prolog's heritage includes the research on theorem provers and other automated deduction systems developed in the 1960s and 1970s. The inference mechanism of Prolog is based upon Robinson's resolution principle (1965) together with mechanisms for extracting answers proposed by Green (1968). These ideas came together forcefully with the advent of linear

resolution procedures. Explicit goal-directed linear resolution procedures, such as those of Kowalski and Kuehner (1971) and Kowalski (1974), gave impetus to the development of a general purpose logic programming system. The other major influence on the nature of this first Prolog was that it was designed to facilitate natural language processing.

2.12 SUMMARY

This chapter is regarding to all the researches that done mainly to gain information for this project. All the information gathered includes the concept and fact on intelligent assessment. Review is also has been done on the existing system and further reading about all the development tools that will be used when developing the project.

Research on intelligent assessment concept is done to have better understanding on the requirement phase of this project.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

Basically, before developing a system, the proper methodology must be identified clearly. A methodology may be defined as a collection of procedures techniques, tools and documentation aids. The procedure, techniques, tools and documentation aids helps software developer to speed up and to simplify the software development process. A methodology also helps the system developer to plan, manage, control and evaluate information system project.

3.2 SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)

SDLC is also an abbreviation for Synchronous Data Link Control.

The systems development life cycle (SDLC) is a conceptual model, used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application. Various SDLC methodologies have been developed to guide the processes involved, including the waterfall model (which was the original SDLC method); rapid application development (RAD); joint application development (JAD); the fountain model; the spiral model; build and fix; and synchronize-and-stabilize. Frequently, several models are combined into some sort of hybrid methodology. Documentation is crucial regardless of the type of model chosen or devised for any application, and is usually done in parallel with

the development process. Some methods work better for specific types of projects, but in the final analysis, the most important factor for the success of a project may be how closely the particular plan was followed.

In general, an SDLC methodology follows the following steps:

- The existing system is evaluated. Deficiencies are identified. This can be done by interviewing users of the system and consulting with support personnel.
- The new system requirements are defined. In particular, the deficiencies in the existing system must be addressed with specific proposals for improvement.
- The proposed system is designed. Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues.
- The new system is developed. The new components and programs must be obtained and installed. Users of the system must be trained in its use, and all aspects of performance must be tested. If necessary, adjustments must be made at this stage.
- The system is put into use. This can be done in various ways. The new system can phased in, according to application or location, and the old system gradually replaced. In some cases, it may be more cost-effective to shut down the old system and implement the new system all at once.

- Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all times. Users of the system should be kept up-to-date concerning the latest modifications and procedures.

3.3 OTHER LIFE CYCLE MODEL

The following paragraphs briefly describe each kind of lifecycle models.

- Waterfall Model

  In the waterfall model software is developed in sequential phases (e.g., requirement, analysis, design, implementation, post delivery maintenance and retirement) with established milestones, documents, and reviews at the end of each phases. There is no overlap between phases.

  While this model provides a useful way to categorize the types of tasks that occur throughout the development life cycle, it does not recognize the characteristics of component-based enterprise development.

  The major problem with the waterfall process model for component-based development is that it is task-focused rather than process-oriented. This makes it difficult to make the flexible

decisions and meet the rapidly changing priorities that are vital to managing an enterprise development project with its multiple components and heavy emphasis on user interface requirements.

The waterfall model is well suited to projects that have stable requirements.

- V-Shape

The V-shaped model is similar to waterfall except it emphasizes the importance of considering the testing activities up front instead of later in the life cycle. The advantage of this model is that test plans are written as requirements elicitation progresses.

- Prototyping

This model is useful in "proof of concept" or situations where requirements and user's needs are unclear or poorly specified. The approach is to construct a quick and dirty partial implementation of the system during or before the requirement phase. The advantages of this model are to allow the user to see and use the proposed solutions and to develop specifications from the prototype. The model helps to collect user feedback to ensure the real products will meet user requirements.

The disadvantage of this model is that the client can think of the prototype as being the final product. In some cases designers spend too much time perfecting the prototype.

- Incremental

This model consists of a series of stages or increments. Each increment adds functionality to the project and each increment has its own complete lifecycle, means each stage is composed of design, code and unit test, integration test and delivery. The increments can be built serially or in parallel depending on the dependencies among the deliverables.

The incremental model has many advantages. Firstly, it allows an easier management of the project because it is simpler to manage each increment separately. Secondly, it also allows putting functional software into the hands of the customer much earlier than either the waterfall or v-shaped model. Finally, it is easier to modify an increment when there is a change to the user requirements. The incremental model is well suited for complex projects and when requirements are not very clear.

❖ Spiral Model

The spiral lifecycle combines elements of the waterfall lifecycle model, and it places more emphasis on the use of risk management techniques. Each spiral identifies potential risks and plans the next step based on the risk analysis. After all the risks have been addressed, the spiral model terminates as a waterfall software life cycle.

The main advantage of this model is obviously the risk assessment. Another advantage is that once risks are well assessed different models can be incorporated to the lifecycle instead of the waterfall model. This model is well suited for complicated projects, with high requirements volatilities.

3.4 CHOOSEN METHADOLOGY

To develop this project I would like to choose Waterfall Model as my methodology that I will follow through out all the processes.

3.4.1 Waterfall Model

The waterfall model is a popular version of the System Development Life Cycle for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

The following diagram gives a visual layout of the waterfall method which will represent the project system development process. Each stage must be completed before moving on to the next. Testing is done in each stage and the developer may go back to the previous stage to correct any error.

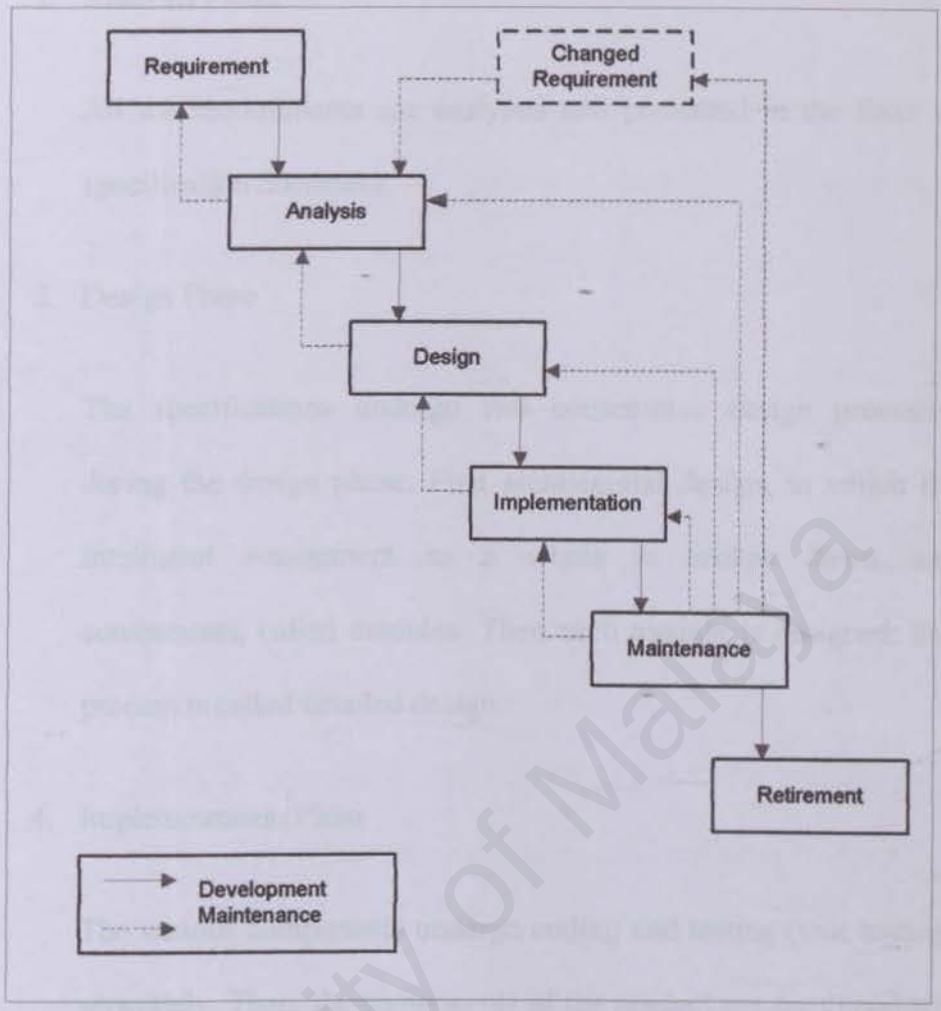


Diagram 3.1 Waterfall Model

The six stages of Waterfall Model

1. Requirements Phase

   The concept of Intelligent Assessment System is explored by finding the knowledge via Internet and met with the supervisor.

2. Analysis Phase

All the requirements are analyzed and presented in the form of specification document.

3. Design Phase

The specifications undergo two consecutive design processes during the design phase. First architectural design, in which the Intelligent Assessment as a whole is broken down into components, called modules. Then each module is designed; this process is called detailed design.

4. Implementation Phase

The various components undergo coding and testing (unit testing) separately. Then, the components of the product are combined and tested as a whole; this is termed integration. When the developers are satisfied that the product functions correctly, it is tested by the client (acceptance testing). The implementation phase ends when the product is accepted by the client and installed on the client's computer.

5. Maintenance Phase

The product is used to perform the task for which it was developing. During this time, it is maintained.

6. Retirement Phase

Retirement occurs when the product is removed from services. This occurs when the functionality provided by the product no longer of any use to the client organization.

There are a few reasons for choosing Waterfall Model as development methodology.

- Waterfall Model is simplicity, which means that it is simple and easy to understand.
- The next Phase only starts when the previous phase has finished. This means that each phase can be fully focused and no need to think the next phase until the current phase has finished.
- Testing is performed in every stage. So it is flexible to go back to the previous if some error occurs or something has been missed out which offers backward feature.

3.5 CHOOSEN TECHNOLOGIES

I have decided to use all the technologies that will be listed below as my specification on developing the Intelligent Assessment System.

3.5.1 Selected Operating System

The Operating System that selected to develop this system is Microsoft Windows XP Professional. Windows XP Professional is the latest version of Microsoft Corporation has even produced after the latest edition of Microsoft Windows Server 2003. One of the key features of Windows XP that make every personal computer where use this platform is more stable and reliable than the previous version (Windows 2000 Professional). Also key enhancement also inherited from Windows 2000 Professional.

1. Application compatibility technologies.
2. Shutdown event tracker
3. Control of unresponsive applications.
4. Windows installer.
5. Auto update, Dynamic update and Windows update.
6. Shadow copy integration
7. Last known good configuration.
8. Automated system recovers.

3.5.2 Selected Programming Language

I have decided to use Visual Basic 6 for the programming language that I will use to develop this system. Visual Basic also provides several significant advantages as listed below:

- VB 6.0 is one of the most popular programming tools in Windows environment due to its RAD (Rapid Application Development) capability that associated with it.
- VB 6.0 is embedded with search engine (JET engine 1.0) that comes from the family that is similar to the internal engine (JET engine 2.0) of DBMS used.
- VB 6.0 uses an event driven approach to program the event driven model response to event that happens in the computer environments. Such events include the pressing of mouse button or call function from another application that running concurrently.

Hence, Visual Basic is the first and the only choice to be use as the development language on this project. Hopefully.

3.6 SUMMARY

The proposed system will be implemented using waterfall model. Thus, the system will be developed through several stages. Through the system development life cycle, system methodology is adopted to understand the current problem situation. The next chapter will discuss on the software and hardware requirements, functional and non-functional requirements for the proposed system.

CHAPTER 4

SYSTEM ANALYSIS

4.1 INTRODUCTION

The fact finding and requirement analysis activities are closely related and are often interleaved (Whitten et al, 2002). If requirements found during the fact finding process seem to be a problem, the analysis activities is done in order to resolve the problems before continuing to elicit additional system needs and desires. The requirement can be categorized as functional requirements and non-functional requirements.

4.2 INFORMATION GATHERING METHODS

- Internet Surfing

  Currently, the internet is a host to numerous materials, including academic information. Internet in the past has proven useful for researchers to communicate with each other, with the availability of electronic mail and discussion groups. Over the years, the internet still remains as a platform for people to inter-network and communicate from all over the world.

  Through the internet, I can collect some idea from the similar system and some interesting web design. Besides, get a lot of information on distributed system, development tools and technologies, programming language ant others information on how to build my project system.

❖ Printed Material

A broad variety of printed material ranges from books, encyclopedias, magazines, newspaper, dictionaries and others is available. All this aid in the analysis of literature published foe a group or target audience. Printed materials are best used as ready references sources.

❖ Discussion

Consistently discussion with supervisor has been practiced from time to time in order to get help and advices during the development of the project.

4.3 FUNCTIONAL REQUIREMENTS

Functional requirements are statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situation. It also describes an interaction between the system and its environment. The following are a number of functional requirements for the proposed learning package:

- ❖ The user can learn several important projects through the system.
- ❖ The user can evaluate their understanding in the topic learn by trying to solve the tutorial question after each chapter.

The following represents the functional requirements for the proposed project. The Intelligent Assessment System is divided into four main modules.

1. Student module

A student module of the user of IAS system stores information on how much the student knows about concepts and relationships which are to be learnt and about the student's level and achievements. It often contains a history of task performance and some detailed representation of the state of an individual's knowledge in a specified subject area.

At minimum, such as a model tracks how well a student is performing on the material that being taught. Student model can be divided into three types of user that will use this system. There is beginner, intermediate and advance student. They are categorized by taking a pretest examination after they login into the system.

2. Learning module

The Learning module is a representation of the knowledge to be imparted, together with the explanatory facility for remedial intervention. It may also contain an error recognition and evaluation feature, and problem-solving model.

This module also contains the information that being taught to the learner. It is a model of the teaching process. For example, information about when to review, when to present a new topic and which topic to present is controlled by this module.

3. Assessment module

The Assessment model arranges teaching strategy, initiates remedial actions and monitors and assesses performance in conjunction with the evaluative function. Problem-generation from the knowledge-base will offer a sequence of problems, adapting the difficulty level on the basis of previous performance,

but will also present new problem classes and review or practice already known items.

Assessment module links a student to the other parts of the system through the user interface and advises the student on the basis of the work done so far. It allows a student to adopt a different route to the solution and provides feedback whenever a value is entered. If the value entered is correct, it assumes the intermediate steps to be well within a student's conceptual knowledge boundaries. If the value is incorrect, then it guides the student in the graded manner as explained below:

- If the student's output is correct

  The assessment module compares the student' output with the correct outcome that derived by the system. If the values match, the student can then proceed further to another question.

- If the student's output is incorrect

  The assessment module provides a progressively graded help.

  i. Ask the student to try again because the value entered is incorrect.

ii. Suggest the help function based on their type of user either beginner, intermediate or advance student.

iii. From the help function, they are given a hint or the suggest data and information that they need to mitigate the question given.

4. The User Interface module

The user interface module is the communicating component of the IAS which controls interaction between the student and the system. In both directions, it translates between the system's internal representation and an interface language that is understandable to the student.

The User Interface module provides interaction between the user and the various part of the system. It includes user driven learning, dialog box and user instruction. It also manages the information that will present to the student effectively.

4.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements as the name suggest are those requirements which are not directly concerned with the specific functions delivered by the system. It describes a restriction on the system that limits our choice to constructing a solution to the problem. These constraints usually narrow our selection of language, platform or implementation techniques or tools. However, the solution is made at the design after the requirements has been specified.

- User Friendliness

  The system is required to have a very user friendly interface because the student will use this system is either expert user or novice user.

- Graphical User Interface

  Graphical User Interface (GUI) is an important part of this project. The project will fulfill the characteristics of GUI which are windows, icons, menus and pointers (WIMP). GUI acts as communicator between the user and the system, making it easy for the users to use the system. Well designed GUIs will instance the usability of the system.

- Security

  The system requires a sufficient security. Users are required to enter their identity and password to access the system and data. This will increase the security level of the system.

- Reliability

  Reliability extends to which a system can be expected to perform its intended function with required precision and accuracy. Thus, the system should be reliable in performing its daily functions and operations correctly.

- Usability

  The final product should be developed in such a way it is easy to use. It will enhance and support rather then limit or restrict the office processes. Human interfaces need to be intuitive and consistent with all other sub-systems.

4.5 RUNNING TIME REQUIREMENTS

4.5.1 Software Requirements

- For the operating system, Windows 2000 and XP Professionals are needed.
- There is any platform with Graphical User Interface (GUI).

4.5.2 Hardware Requirements

- Pentium III 250 MHz and above is required.
- 128MB RAM is recommended.
- CD-ROM, keyboard and mouse are required.
- Others standard computer peripherals.

4.6 SUMMARY

System analysis is one of the most important phases in software development. Requirement analysis helps determine the software components that required accomplishing the required function of the system.

The functional and non-functional requirements are given a clear an unambiguous picture of the system functionalities and constraints. The information gathering methods that have been done are to get some general knowledge and information. Besides that, minimum run-time requirements for both client and server side are determined in this chapter.

By having all these techniques, the users' requirements will be met and the system will be built out nicely.

CHAPTER 5

SYSTEM DESIGN

5.1 INTRODUCTION

Design is the creative process of transforming the problem into a solution, the description of a solution is also called design (Pleeger, 1998).

A system model is a representation of an in-place or proposed system that describes the data flow throughout the structure. The model describes the points where data or information enters a system and the process where it will be processed, as well as the actions taken and the points where the data will be output.

Design diagrams include data flow diagram (DFD), structured charts, decision trees, and other items. DFD shows how the data flow into the system, how there are transformed and how they leave the system. It also depicts the function that transform.

The common steps involved including analyzing, designing, coding and testing the system to ensure that it conforms to the software specifications and requirements. The design of this system can be viewed from the following aspects:

- System functionality design
- GUI design

5.2 SYSYEM FUNCTIONALITY DESIGN

In the system functionality design, it will include description on the system structure chart, data flow diagram and context-flow diagram for the Intelligent Assessment System.

5.2.1 SYSTEM STRUCTURE CHART

The main modules of the Intelligent Assessment System are portrayed as labeled rectangles in the structure chart. Modules are factored into sub modules through the top-down approach. The project structure chart is resultant of the comprehensive and analytical study of the flow of the system modules. The following illustrated the modules involved.

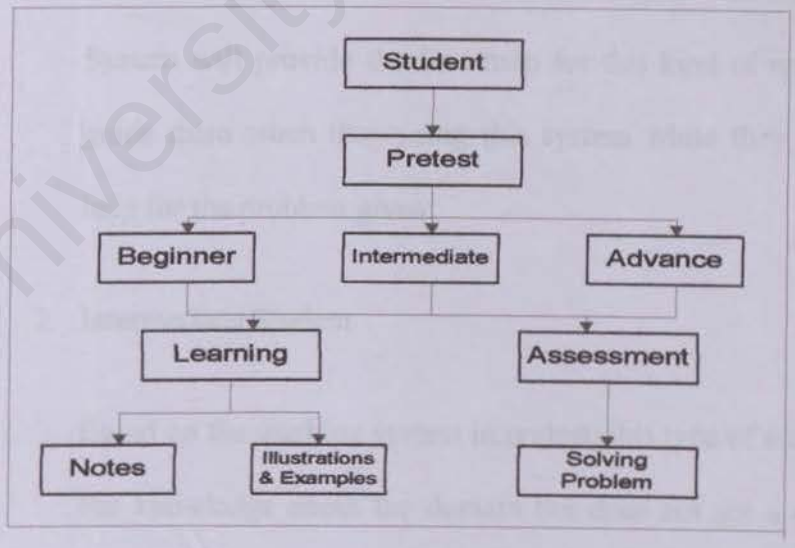


Figure 5.1 The Intelligent Assessment structure chart

5.2.2 STUDENT MODULE DESIGN

Student module is the basic component of the system. It is divided into three major types of user that will be using this system. Functionality of each type will be explained later. Before the student is categorized by their types, they need to seat for a pretest examination to identify which type they will be based on the system marking.

Three Types of user are

1. Beginner Student

   Based on the marking system in pretest, this type of student does not have any knowledge in robotic and know very little about the robotic domain. Intelligent Assessment System will provide the first help for this kind of user to guide them when they using this system while they need help for the problem given.

2. Intermediate Student

   Based on the marking system in pretest, this type of student has knowledge about the domain but does not get a clear picture when solving the tutorial problems. Intelligent

Assessment System will provide the second help for this kind of user.

3. Advance Student

This type of student has advance knowledge about the domain. The system will provides the third help for helping them in solving the problems.

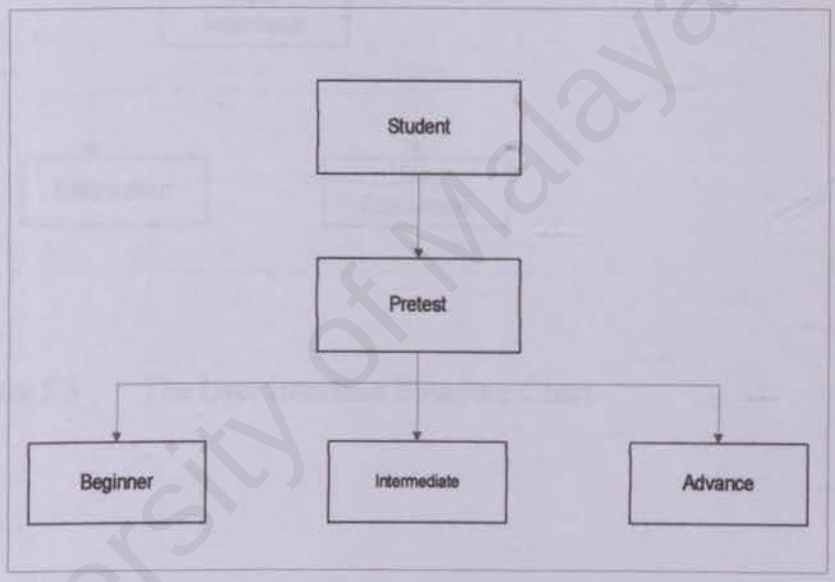


Figure 5.2 Student Module Structure Chart

5.2.3 The User Interface Module Design

The functionality of this module is to process flow communication direction between the student and the system. This module contains the instruction and user guidance to help the student to go through the various parts in this system.

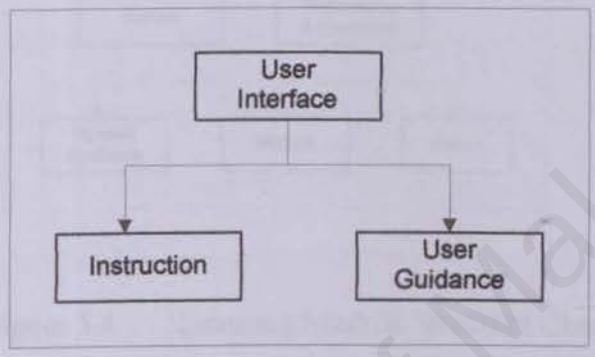


Figure 5.3 The User Interface Structure Chart

5.2.4 Learning Module

This module come from the knowledge base that contains notes and illustration and examples for the student to view all the learning notes that they want to learn before they take for an assessment. For the Intelligent Assessment System, three topics in robotic are integrated. They are speech synthesis, motion and

vision. Student is optional to select the topic that they really want to learn.

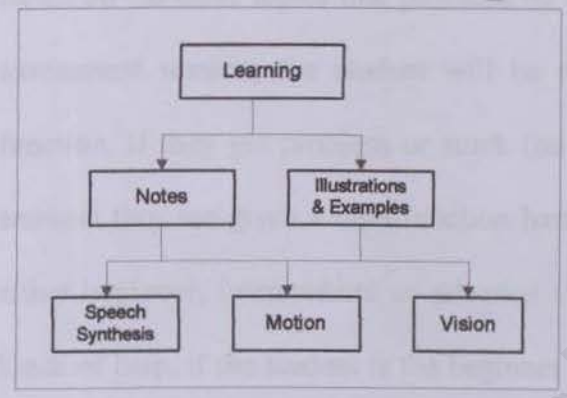


Figure 5.4 Learning Module Structure Chart

5.2.5 Assessment Module Design

This module contains the problem solving sub module. Student is freely to choose which topic they want to take as an assessment based on the three topics that provided by this system. During the assessment session, the student will be provided with the help function. If they get problem or stuck (no idea) when giving the answers they are given a help function based on their type of user either beginner, intermediate or advance student. There are three kinds of help, if the student is the beginner student the first help is available for him, meanwhile the other two helps will disable for him to retrieve hint from that help. The process is the same for intermediate and advance student.

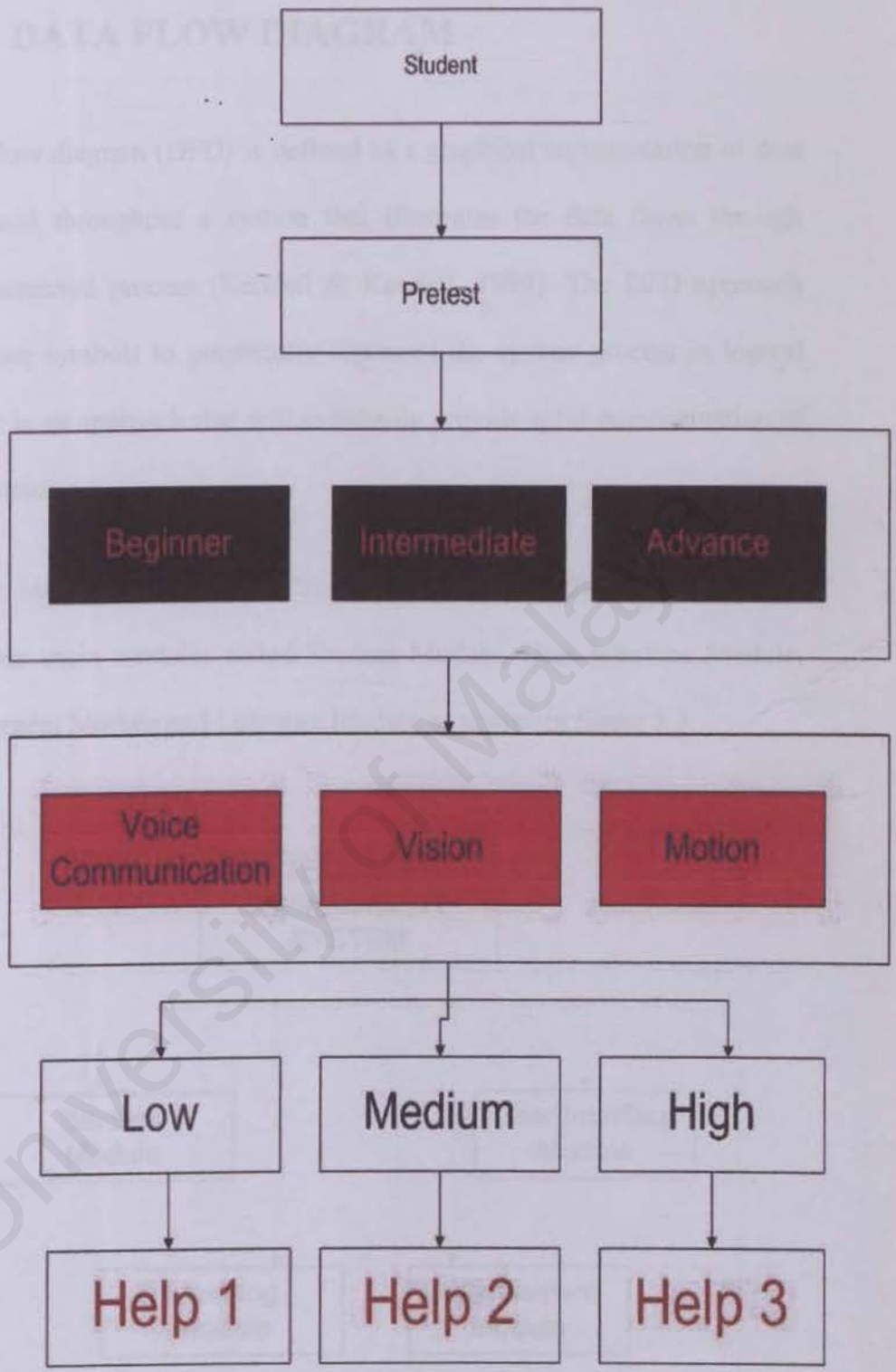


Figure 5.5 Assessment Module Structure Chart

5.2.6 DATA FLOW DIAGRAM

Data flow diagram (DFD) is defined as a graphical representation of data processed throughput a system that illustrates the data flows through interconnected process (Kendall & Kendall, 1999). The DFD approach uses four symbols to graphically represent the system process in logical way. It is an approach that will eventually provide solid documentation of the system.

For the Intelligent Assessment System, the system architecture is divided into four main modules called Student Module, User Interface Module, Assessment Module and Learning Module as shown in figure 5.2

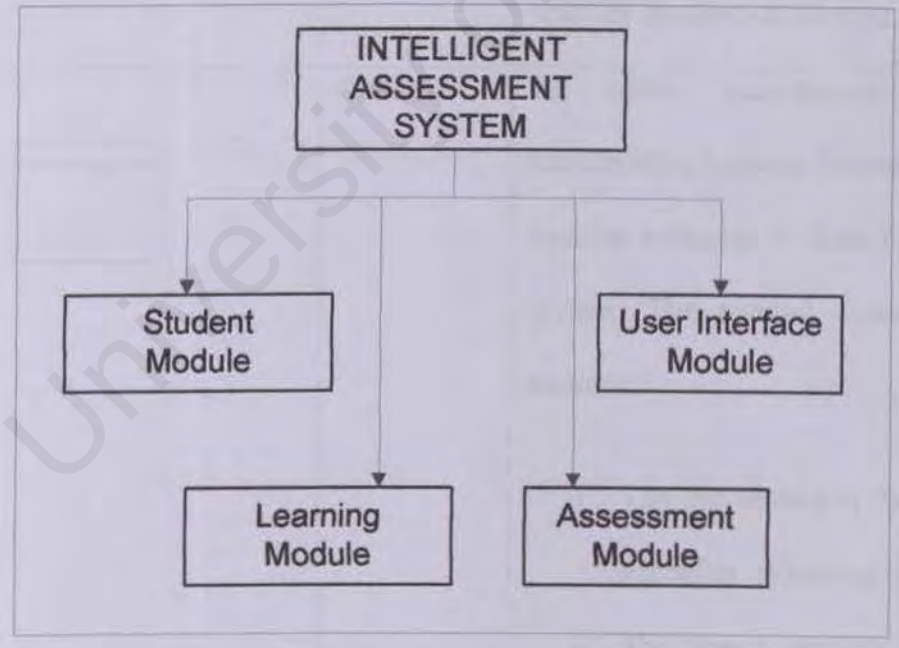
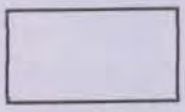
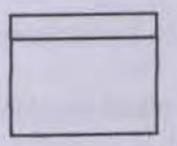


Figure 5.6 Intelligent Assessment System and its main Modules

Symbol	Meaning	Description
	Entity	To depict an external entity that can send data to or receive data from the system. Also known as source or destination of data and considered beyond the boundaries of the system
	Flow of data	To represent the flow of data or information from one point to another. The arrow describes the directions of the flow, with the arrowhead pointing to the data destination. Each data flow is labeled with the detailed of the data.
	Process	To show occurrences of a transforming process. Process always denotes a change in data within the system. The symbol consist of 2 sections: 1. The top section is the unique identifier indicating its level. 2. The lower section contains the description of the process.

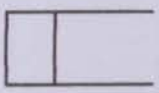
	Data Store	To represent data store and holds data for a given time within the system. The symbol consists of 2 section: 1. Identifier reference number 2. Description of the data stored

Table 5.1 Descriptions of the conventions used in Data Flow Diagram

Context Data Flow Diagram

A context data flow diagram (Whitten et al, 2002) defines the scope and boundary for the system and project. The context diagram below is constructed to establish an overview of the initial project scope which includes basic inputs, the general system and outputs.

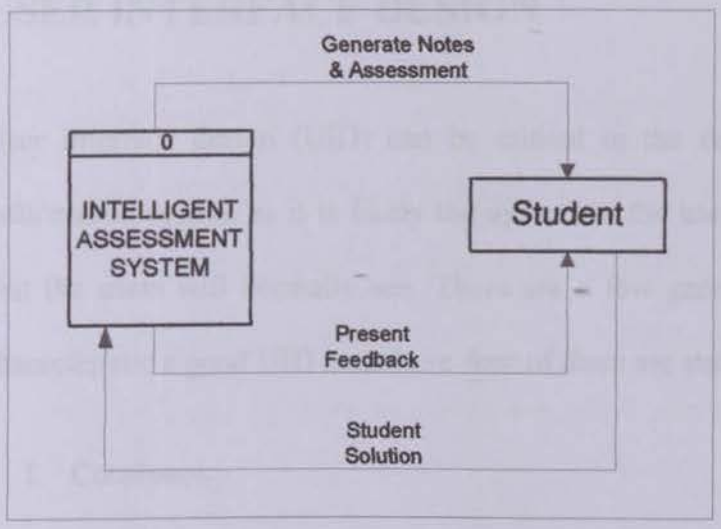


Figure 5.7 The context Data Flow Diagram of Intelligent Assessment System

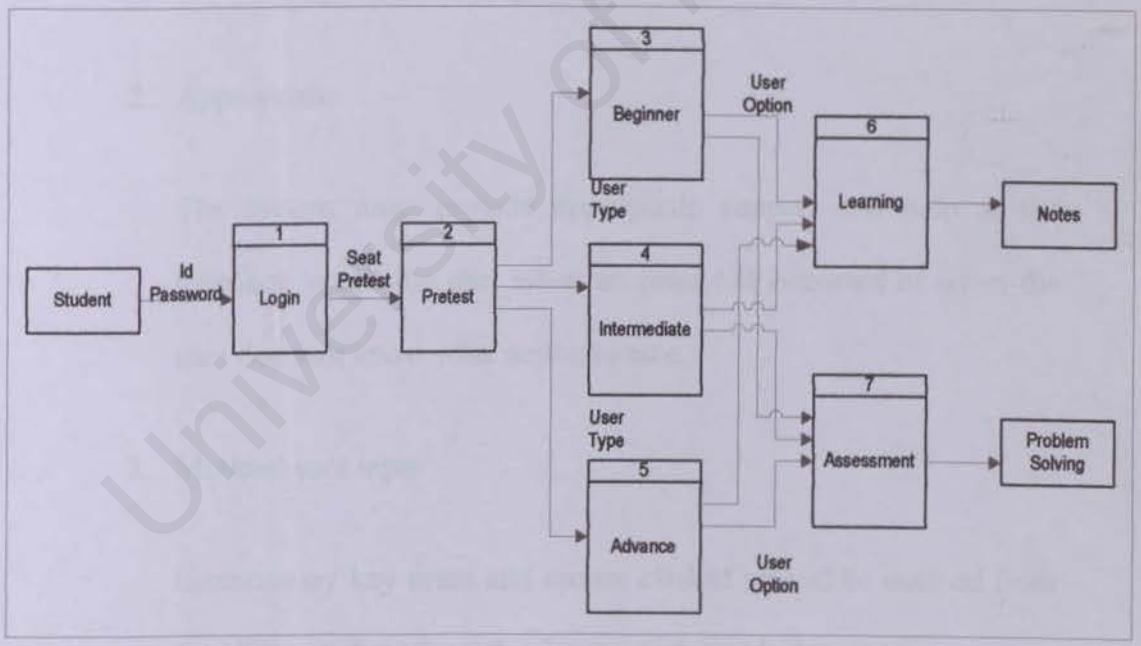


Figure 5.8 DFD for Intelligent Assessment System

5.3 USER INTERFACE DESIGN

User Interface design (UID) can be critical in the development of an information system as it is likely the system to the user because it is all that the users will normally see. There are a few general but important characteristic a good UID must have four of them are stated below:

1. Consistency

   Helps users to learn an application and to apply what they know to different parts of a system easily. This includes the commands, format of data entry and layout of the interface. This aids the user in learning process as the interfaces are consistent.

2. Appropriate

   The system must provide appropriate support and help at the interface to tell the user when an error has occurred or when the user does not know what action to take.

3. Minimal user input

   Unnecessary key press and mouse clicked should be omitted from the UID to reduce the risk of errors and speeds data entry.

4. Adequate feedback from system

User expects the system to respond when they take some action. This is to show the user whether the system has noticed then action. If not the user can take that action again, thus notifying the user of the status of their action. The system's responses time should be appropriate as a long response will create confusion among users.

5.2.1 PROTOTYPING

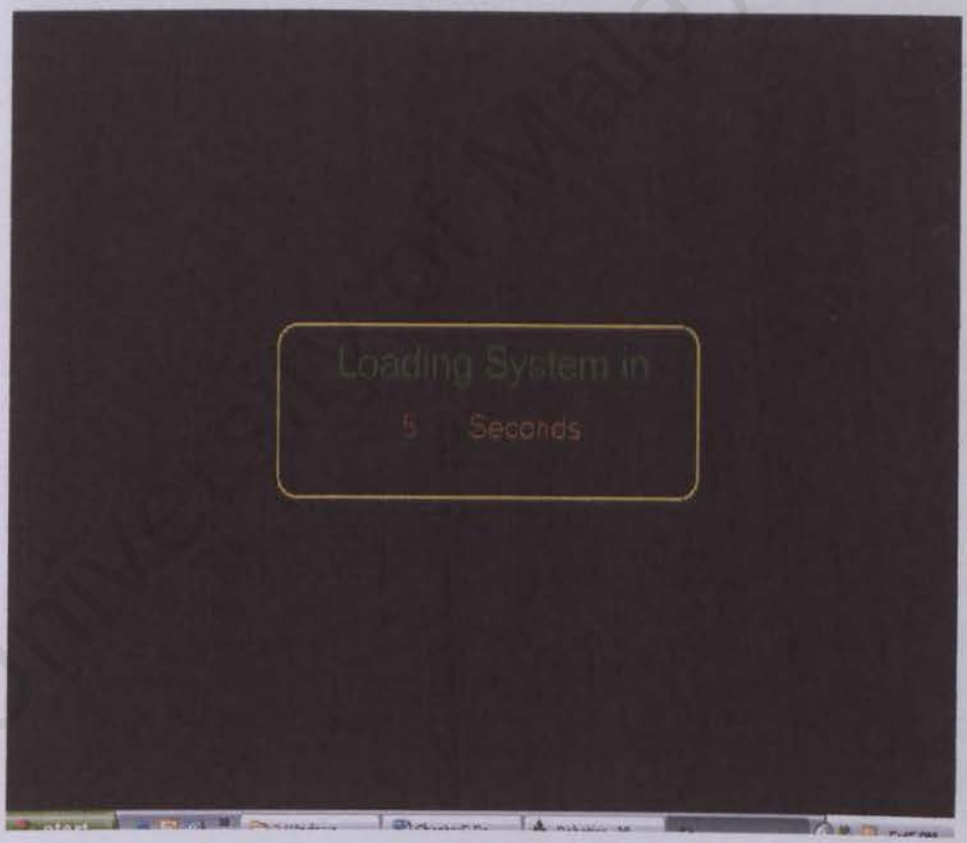


Figure 5.9 The main interface for Intelligent Assessment.

Figure 5.10 Pretest Form

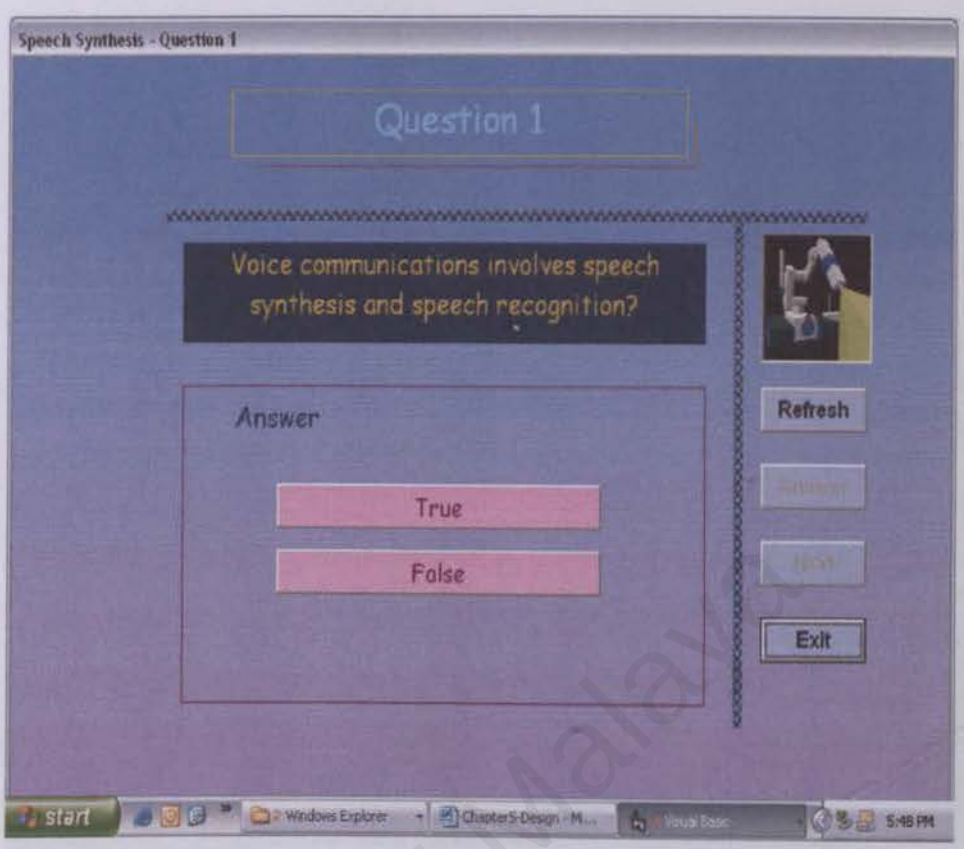


Figure 5.11 Sample of Pretest Question

Figure 5.12 Result Form

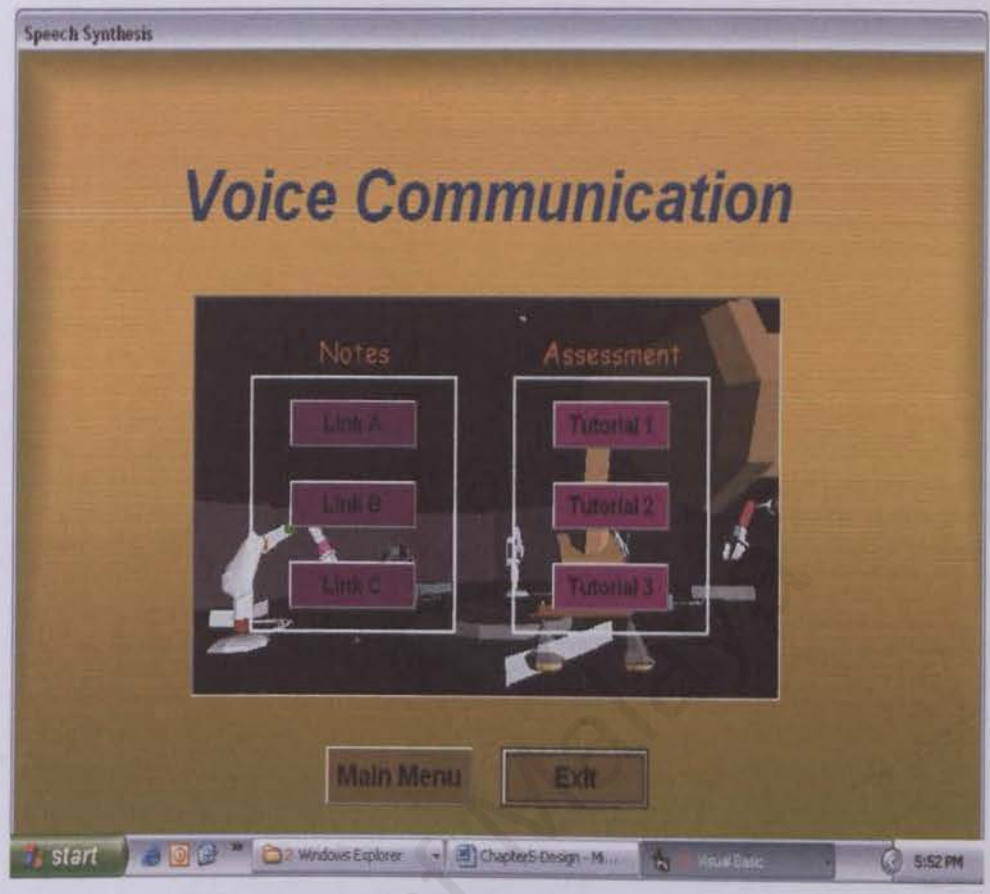


Figure 5.13 Voice Communication Form

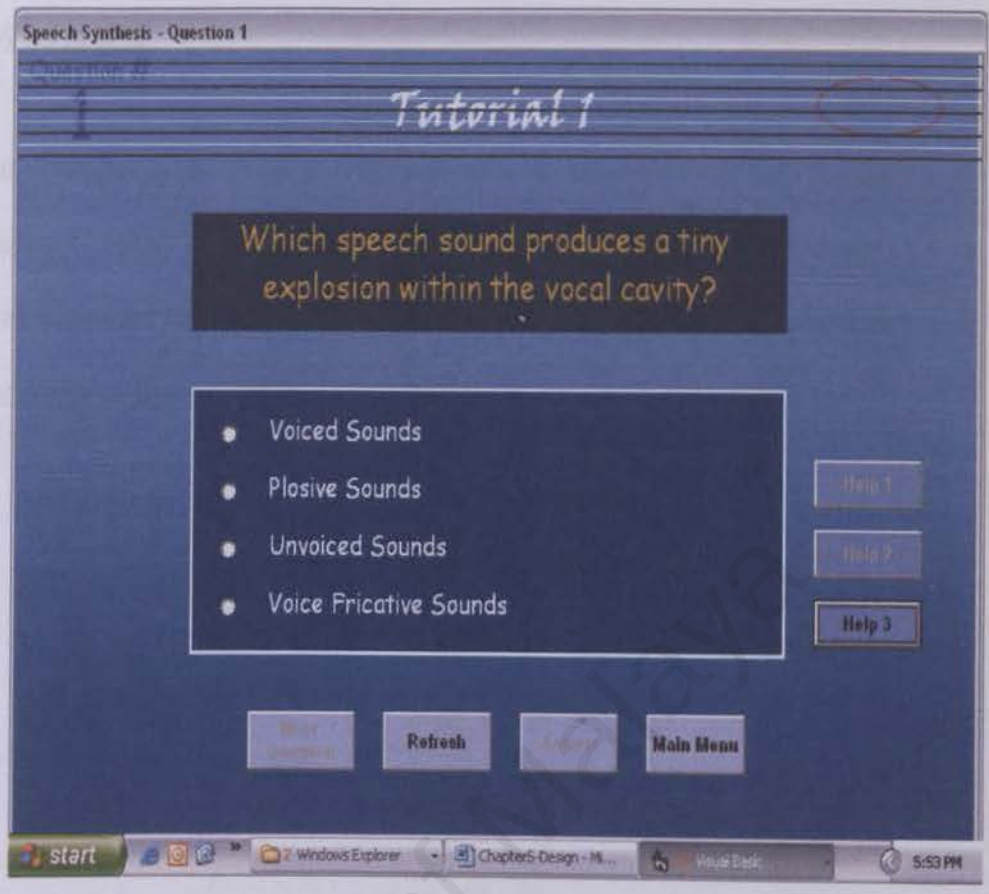


Figure 5.1.4 Form of Question 1 that shows Help 3 is enabling for the intermediate student based on their categorized result which is high result.

5.4 SUMMARY

System design is a process of converts a conceptual idea in the requirement specification in the system analysis into the real design contains features, component and elements. The explanation about user interface design is also stated and prototyping.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 INTRODUCTION

This chapter discuss about the steps and methods taken in implementing earlier proposed design unto developing the system. The developed system will be tested for functionality and errors.

System implementation is the acquisition and integration of the physical and conceptual resources that produces a working system. It is this physical realization of the application design. System implementation includes building and testing its contained modules and sub-modules, involving system requirement and design conversion into program code.

6.2 DEVELOPING ENVIRONMENT

The initial stage of system implementation involves setting up the development. The usage of dynamic and suitable hardware and software could help accelerate the development or construction of the system. The hardware and software tools used to develop the entire system are discussed below:

6.2.1 HARDWARE TOOLS

The hardware used to develop the system is as listed below.

- Pentium III 250 MHz and above
- 128MB RAM

- ❖ Other standard computer peripherals

6.2.2 SOFTWARE TOOLS

The whole system is developed using Microsoft Visual Basic 6.Visual Basic 6 is powerful tools in developing windows application. The main reason is the suggestion from my supervisor and my lack of knowledge in other windows programming. Visual Basic 6 also includes much functionality such as debugging which makes it easier to find errors and trace memory usage.

6.3 APPLICATION DEVELOPMENT

Application development involves code generation that translates all the algorithms and design into Visual Basic. Visual Basic is a visual and event driven programming language.

In traditional programming languages, programming is done in test environment and the program is executed sequentially. In Visual Basic, programming is done in a graphical environment. Each sub-program is triggered when the use of the system invokes the events.

Several programming principles have been employed in writing the program to ensure consistency, maintainability and reliability. All the programming principles are as follows:

- Choosing meaningful variable names, procedure, names and form names helps a program to be self documenting without excessive use of commonly.
- All declarations a placed at the beginning of sub-routines of procedure and declarations are separated from the executable statements in that procedure to make the declarations stand out and contribute to program readability.
- Insert comments to document the program and for better understanding when referring back to the codes and also to improves codes readability.

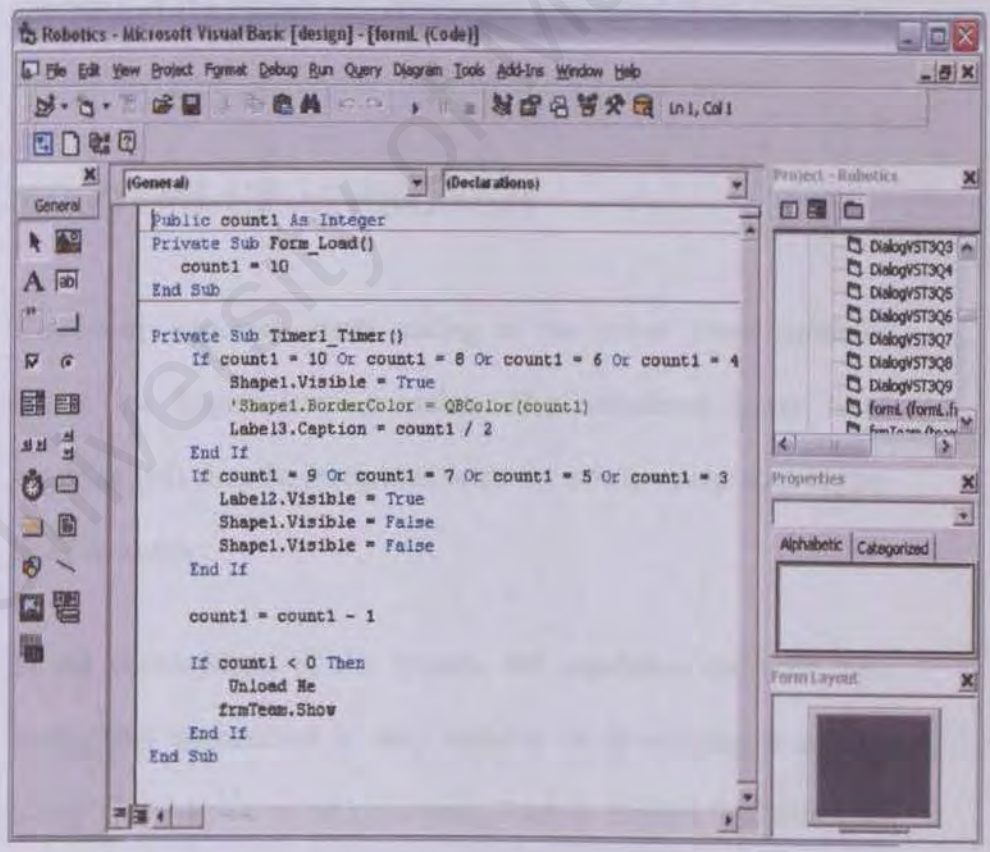


Figure 6.1 The sample of coding

6.4 CODING METHODOLOGY

Basically there are two aspect of coding methodology: there are top-down approach and bottom-up approach.

6.4.1 TOP-DOWN APPROACH

This approach starts by looking at the large picture of the system and then exploding to smaller parts or subsystem. Top-down approach allows the higher level modules to be coded first before the lower modules. This method ensures that important or core modules of the system are developed and tested first. Deploying the method gives preliminary version of the system sooner.

6.4.2 BOTTOM-UP APPROACH

Bottom-up approach starts coding at the lower levels modules before the higher level modules. The completed lower level modules will then be integrated with the newly completed higher level modules.

In the development of the system, the top-down approach was used. This approached is very suitable in developing a system using Visual Basic in which a main form is created first followed by the sub forms.

6.5 SUMMARY

This chapter explains the system implementation which is regarding the transformation of the designed modules and algorithm into the executable instructing by using the appropriate programming language and that is the Microsoft Visual Basic 6. The system testing will be followed in the next chapter.

CHAPTER 7

SYSTEM TESTING

7.1 INTRODUCTION

Testing is one of the main phases in the development of the system. In this phase, the process of testing is debugging are done to detect defects and faults in the system. The process is usually done incrementally with the system development.

7.2 TESTING OBJECTIVES

The objectives of the system testing are as follows:

- To identify, isolate, and correct as many bugs as possible. Most programs have bugs the must inside of which appears only unique combination of data or event.
- To make sure that the functionality of the system is being met.

7.3 TEST CASE DESIGN

Any system can be tested using one of the two types of test case design. There are white box testing and black box testing.

7.3.1 WHITE BOX TESTING

White box testing is also known as glass-box testing technique which uses the control structure of the procedural design to derive

test cases. By using this technique the developer can ensure (Pressman, 2000).

- ❖ All independent paths within a module have been exercised at least once.
- ❖ Exercise all logical decisions on their true and false sided.
- ❖ Execute all loops at their boundaries and within their operational bounds.
- ❖ Exercise internal data structures to ensure their validity.

The main purpose of this technique is to ensure that all detail and often-left unnoticed errors are taken care off. Furthermore, it also helps to ensure the developed system is operated according to the specification stated earlier.

7.3.2 BLACK BOX TESTING

Black box testing is focuses on the functional requirements software. This testing technique also enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of the system. It used to show that the software functions are operational concisely. The objectives of black box testing are to uncover errors in

- ❖ Incorrect or missing function
- ❖ Interface error
- ❖ Performance errors
- ❖ Initialization and termination errors (Pressman, 2001)

Black-box testing is not an alternative to white-box technique. It compliments white-box testing and is likely to uncover a different class of errors.

Black-box testing usually to be applied during later stages of testing which is different from the white-box testing which is performed early in the testing process.

7.4 TESTING STRATEGIES

There are various types of testing available to assess the completeness and correctness. The testing strategy for the system is divided into three main tests: unit testing, integration testing and system testing.

7.4.1 UNIT TESTING

In the unit testing, each component is tested individually. All the important control paths in the project are tested to uncover errors within boundary of the modules. There are three kinds of testing strategy carried out for the unit testing. There are code review,

compilation and test cases. All of these activities were carried out together in a time.

- Code review

  The codes are examined line by line to ensure that any uncovered semantic errors during the implementation could be revealed. There are two types of code review á walking through and an inspection.

  In a walkthrough, the codes and documentation will be presented then the comment on their correctness will be collected. During inspection, the testers check the code and documentation against a prepared list of concerns. Review on the algorithm and computations will be taken to ensure their correctness and efficiency.

- Compilation

  Next, when compilations of codes are done, it will eliminate remaining syntax faults.

- Test cases

  After reviewing and compiling the cases, text cases are developed to test that the input is properly converted to the desired output. To test a component, some input data and

conditions are chosen and then allow the component to manipulate the data and observe the output. In the unit testing, all the independent components are tested and make sure all the outputs are correct and meet requirements.

7.4.2 INTERGRATION TESTING

When the individual modules and components are working properly and meets the objectives, these modules are integrated into a working system. Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with the interfacing. The objective is to take unit tested components and build a program structure that had been dictated by the system.

7.4.3 SYSTEM TESTING

System testing is a series of different test designed to fully exercise the software system to uncover its limitations and measure its capabilities. The objective is to test an integrated system and verify that it meets specified requirements. Although each test in the project has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions. There are several type of system testing that are

worthwhile for a system. For this project, two types of system testing are used.

7.4.3.1 FUNCTION TESTING

System testing begins with function testing which focuses on the system functionalities. Each function can be associated with the components of the system that accomplish it. Function testing compares the system actual performance with its requirements, so that test cases for function testing are developed from the requirements document.

7.4.3.2 PERFORMANCE TESTING

The purpose of the testing is to test the run time performance of software within the context of an integrated system. It requires both hardware and software instrumentation. Resource utilization is measured in an exacting fashion. Performance testing addresses the non-functional requirements of the system after all the function testing has been completed. System performance is measured using performance objectives set by potential users as highlighted in the non-functional requirements section as guidelines.

7.4.3.3 ACCEPTANCE TESTING

Acceptance test ensures the users that the system they requested is the system that was built for them. The system built should meet the criteria and requirements of their needs.

7.4.3.4 INSTALLATION TESTING

Installation testing is a final installation. Test is run to allow the user to exercise system functions and documents additional problems that result from being at the actual size.

7.5 SUMMARY

Software testing is an essential element of software quality assurance and represents the review of specification, design and coding. Testing is essential especially to check if there is any bug or error in the system. After the testing, the system should be able to perform correctly and should be free of errors. The next chapter is about system evaluation and conclusion and discusses about the problem faced during the system development and also the significance of the sy

CHAPTER 8

SYSTEM EVALUATION & CONCLUSSION

8.1 INTRODUCTION

Throughout the system developments problems are encountered and most of them were resolved eventually. The system was evaluated through system testing to identify its strength, weakness and limitations and proposals were made for faster enhancement.

8.2 PROBLEMS ENCOUNTERED AND SOLUTION

The following are the major problems encountered from the beginning of the project through the end of the system development process.

1. Lack of knowledge in the language and tools chosen

   Due to the time constraint, the learning and developing process was done in parallel. The Visual Basic 6 is a programming language that was never taught and learnt before. Most of the problems faced were manageable through the references from books and manuals from the Internet and also with a more efficient way such as trial and error.

2. Difficulties in designing the user interface

   Problem that was faced during the development of the system are the lack of knowledge and experience of the real system flow and layout of standard user interface. Therefore it is difficult to design

the appropriate logic and user friendly interface. To get knowledge of the system flow and user interface design some real commercial computer aided learning packages were taken as a references.

3. Difficulties in collecting algorithm

   Due to the time constraint also the process of information and question on robotic must be fast but there was lack of information especially the latest notes on robotics tutorial.

4. Difficulty in defining the term "intelligence"

   Since the term intelligence is so wide and abstract in AI field, it is quite hard to apply this concept into this system. However, based on my opinion and with my supervisor's help and the article that I have gathered, I thought I still able to apply the intelligence concept although it is not 100%.

8.3 STRENGTH AND SIGNIFICANCE OF THE SYSTEM

1. Attractive and Friendly User Interface

   The user interface for the system is attractive and simple where the graphical user interface features have been integrated in the system. All of the functions and links were presented to the user clearly and consistency.

2. Easy to Use

   The system can be considered easy to use for several reasons. First it is because all the buttons and links in the system are clear, readable and understandable. Users of the system will not have problems understanding the system.

3. Security Features

   The security features are taken into consideration for the system as to prevent any unauthorized users from manipulating all the information.

8.4 FUTURE ENHANCEMENTS

Many new and good ideas come about while the system was being developed. Because of the time constraints and due to some features, not all of the ides could be incorporated into the system. It is hoped that the following aspects would be considered in future.

1. Improve in Interface design

   More complex graphics and animations graphics can be incorporated into the system. The functionality of the system should also be improved.

2. Add In More Information

   The information provided in the system can be enhanced in the future. The notes and tutorial on each chapter should be updated and improved.

8.6 SUMMARY

Overall, this project has achieved and fulfilled the objectives and requirements as a tutoring system that provides knowledge and academic assistance to the students. Throughout the development of the system, a lot of knowledge wee gained such as knowledge on the programming language, Visual Basic and programming concepts. All the problems faced and experienced in developing the system should be useful in the future endeavor. It is hope that the system will be taken as a platform and a foundation for more comprehensive and innovative system in the future.

REFERENCES

REFERENCES

1. Patel, Ashok (1998).A Computer Based Intelligent Assessment System For Numeric Disciplines. *Information Services and Use*. 18(1-2). 53-63.
2. Rui, Min Shen, Yi, Yang Tang, Tong, Zhen Zhang (2001). The Intelligent Assessment System in Web Based Distance Learning Education. *ASEE/IEEE Frontiers in Education Conference*. October 10-13, 2001 Reno, NV.
3. Zhou, Yujian and Martha W. Evens.**A Practical Student Model in an Intelligent Tutoring System.** *Proceedings of the Eleventh IEEE International Conference on Tools with Artificial Intelligence, Chicago (ICTAI-99)*, pp. 13-18.
4. Ashok Patel and Kinshuk. (1997). Intelligent Tutoring Tools in a Computer-Integrated Learning Environment for Introductory Numeric Disciplines. *Innovations in Education and Training International Journal, 34(3), pp200-207 (ISSN 1355-8005).*
5. Anderson, 1985. J.R. Anderson, C. F. Boyle, G. Yost, 1985. The Geometry Tutor. In: Proceeding of International Joint Conference on Artificial Intelligence. 1985.
6. Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive tutors: lessons learned. *The Journal of the Learning Sciences,* 4(2), 167-207.

7. Coiffet, P & Cherouze. 1983. An introduction to robot technology.

8. Foong, Soon Kwai. 1999/2000. Robotic Arm Controller (Robotic Arm simulator Module). UM.

9. Hall, E.L & Hall, B.C. 1985. Robotics: A User Friendly Introduction. CBS College.

10. Iqbal Tabani & Akbar Muntasir. 1987. Robot Motion and Task Planning; Simulation and programming of a Robot Arm. Marcel Dekker, Wangshiton D.C

11. Tan, Chee Hoi. 1999/2000. Robotics Arm Controller (Arm Controller Module).

12. Wolovich, W.A. 1987. Robotics: Basic Analysis And Design. CBS College.

13. Aitken, Peter G. 1998. Visual Basic 6 Programming. Blue Book. United States.

14. Gary B. Shelly. 1999.Microsoft Visual Basic 6: Complete Concepts and Techniques. United States.

15. Stephen R. Schach. 2002. Object Oriented And Classical Software Engineering. Mc Graw Hill.

18. http://www.cs.iit.edu/~circsim/documents/ghdiss.pdf (22/06/2004)

19. www.ei.sanken.osaka-u.ac.jp/english/ies.html (22/06/2004)

20. http://ifets.ieee.org/periodical/vol_4_2000/virvou.html (23/06/2004)

21. http://www.sasked.gov.sk.ca/docs/evergrn.html (30/06/2004)

22. http://aplusix.imag.fr (04/07/2004)

23. http://www.icme-organisers.dk/tsg09/HamidChaachoua.doc (04/07/2004)

24. http://www.pitt.edu/~vanlehn/andes.html (04/07/2004)

25. http://www.lrdc.pitt.edu/ (10/07/2004)

USER MANUAL

APPENDIX

USER MANUAL

USER MANUAL

1.1 INTRODUCTION

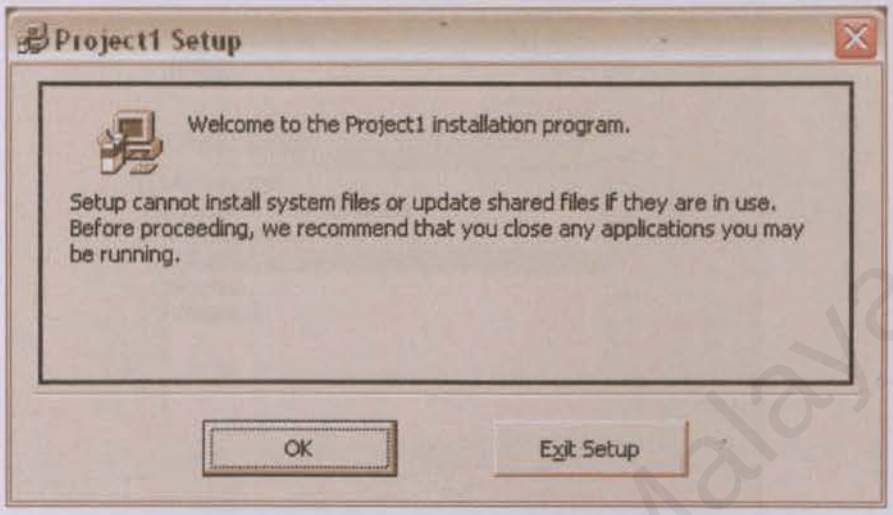
This is the user manual that provide to the users to help navigate through the system. Every major function of will be described in this manual.

The Intelligent Assessment System is a stand alone system that teaches the subject of robotic that includes three chapter of fundamental of Robotics. There are voice communication, vision and motion. This system is integrated with other module which is the knowledge base module. The major module of this system is student module, the user interface module, learning module and assessment module.

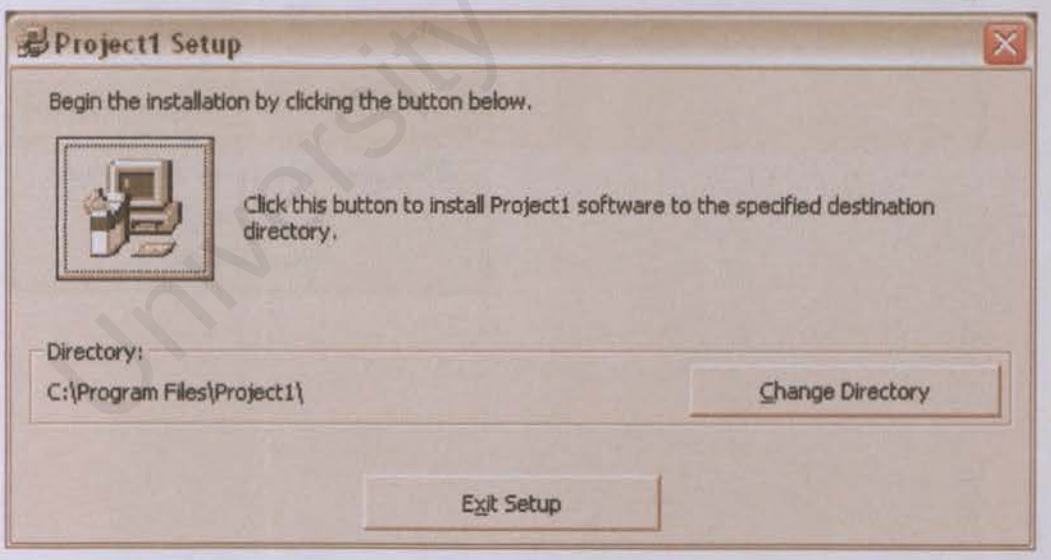
The most focuses is on the tutorial module which is the intelligence of this system is applied.

1.2 SETUP INSTALLATION

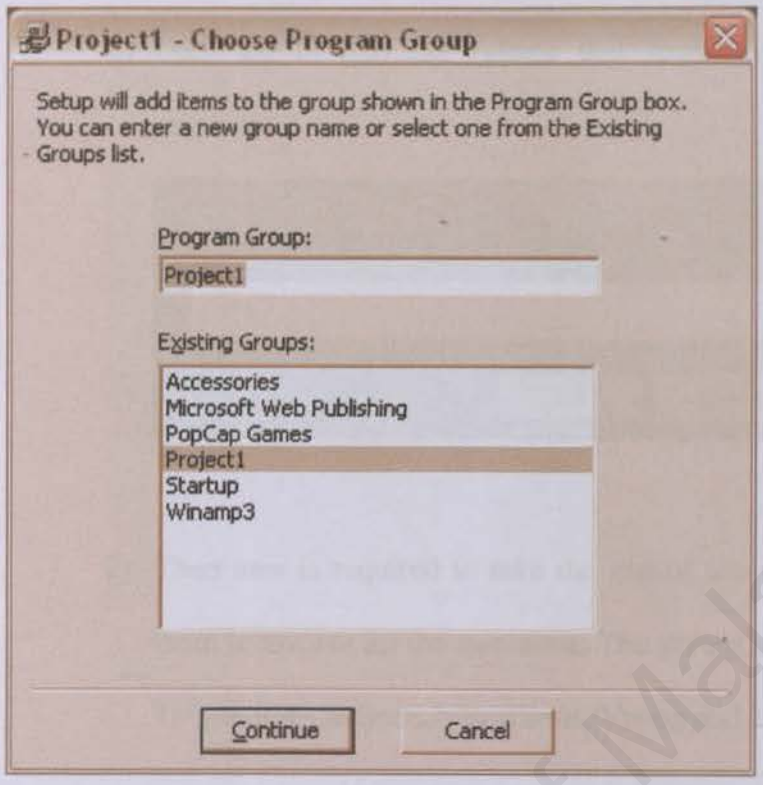
1) Run the setup application



2) Begin the installation



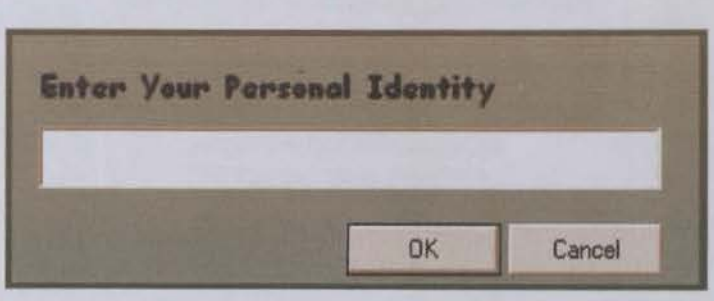
3) Choose Program Group



4} Setup completely successfully

1.3 GETTING STARTED

1) User log in into the system that required user name and password.



2) Then user is required to take the pretest session that required them to answer all the questions. The pretest session is consist 15 questions in Speech Synthesis, Vision and Motion.

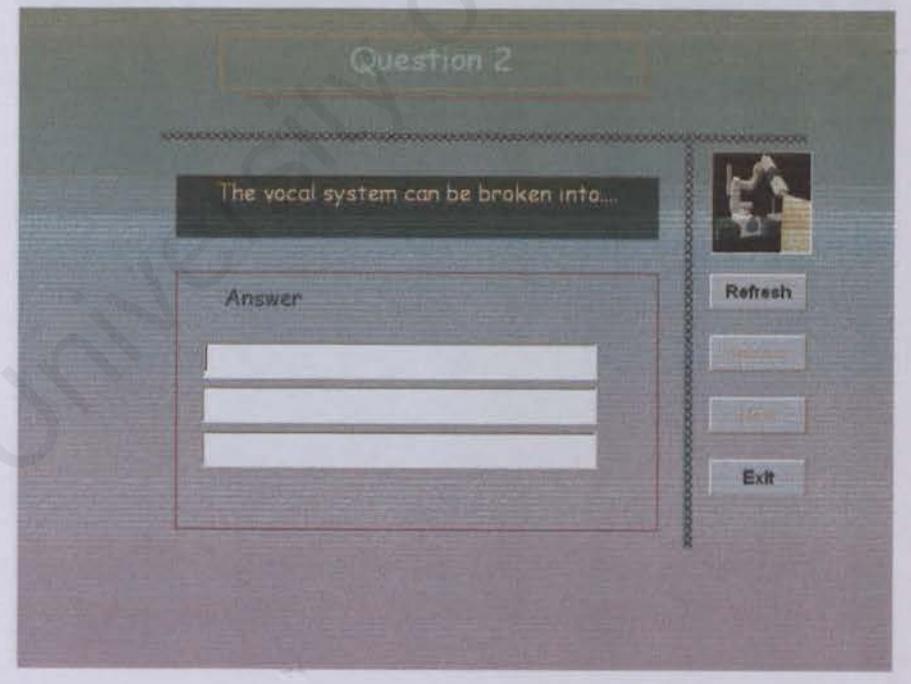


Figure 1.2 Question for speech Synthesis

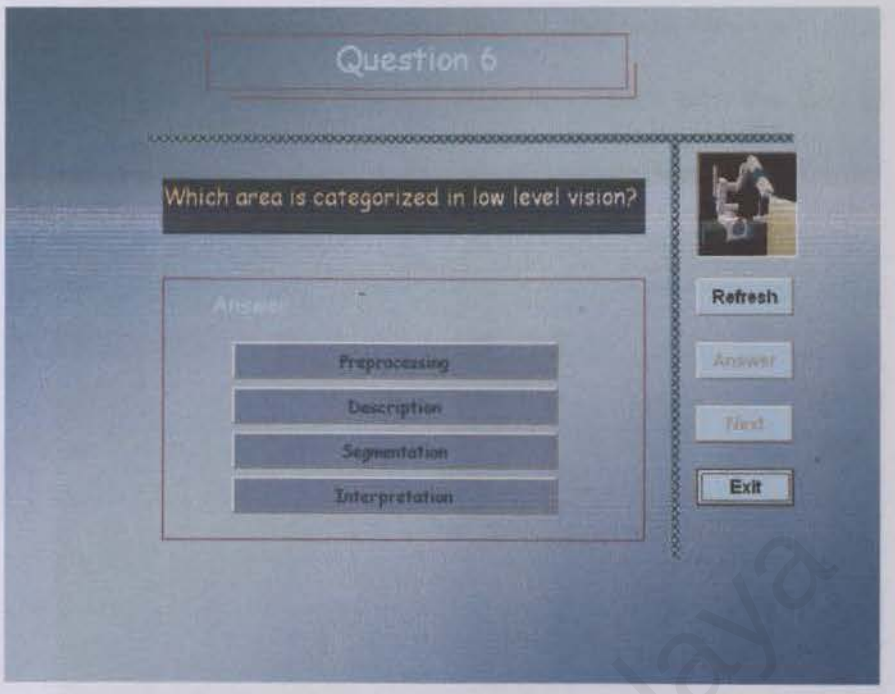


Figure 1.3 Question for Vision

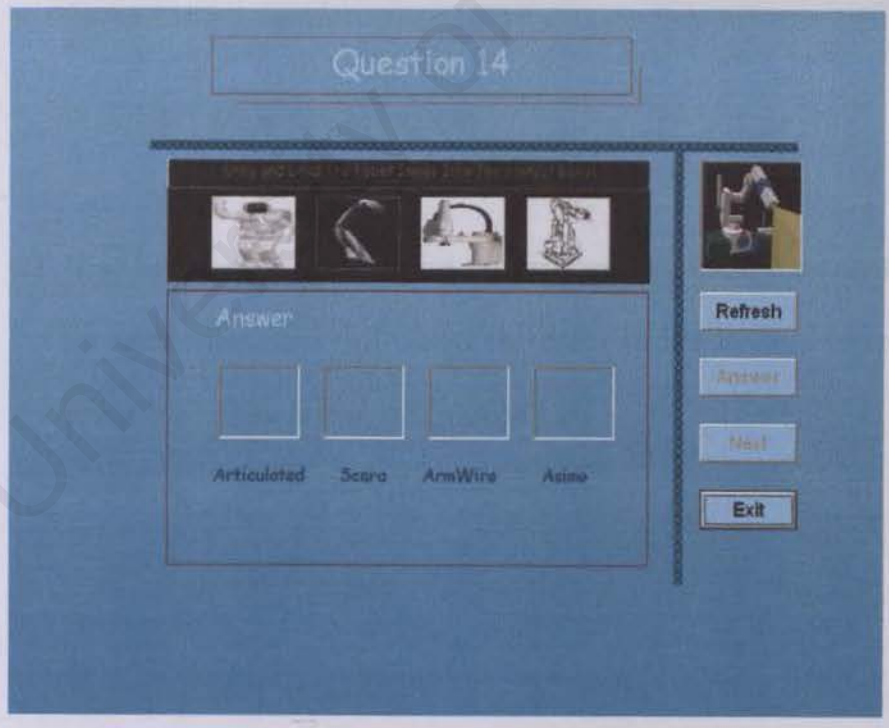


Figure 1.4 Question for Motion

3) After their finished taken the pretest, the user can view their score and from that they are identified which type the user is either beginner, intermediate and advance student.

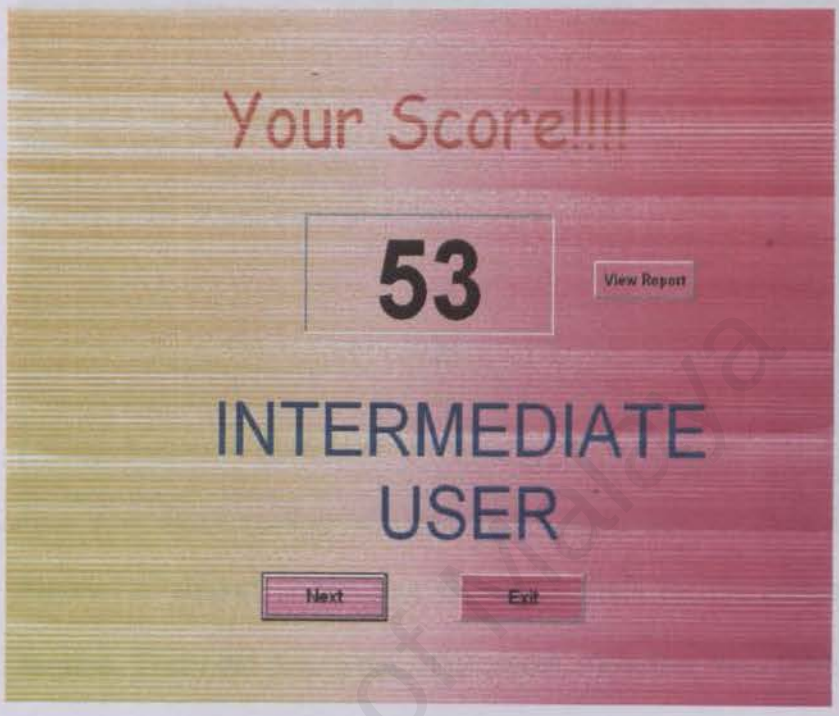


Figure 1.5 Score for the student

4) User can also view the report for their pretest. From the result for each topic, user will know the exact ability in each topic either their mark is low, medium or high.

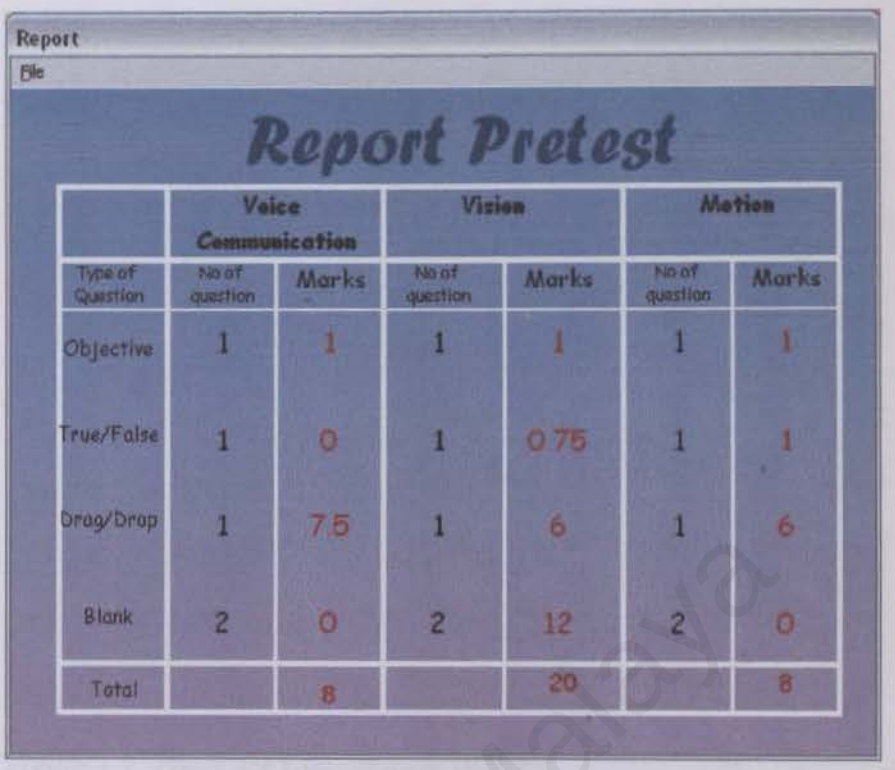


Figure 1.6 Report pretest that shows the mark for the pretest session. There are four types of question, objective, true false, drag drop and fill in the blank questions.

5) User now can click the next button in a score form to go through the system. The system then will display the topic that the student can go through. User can select any topic they are interested.

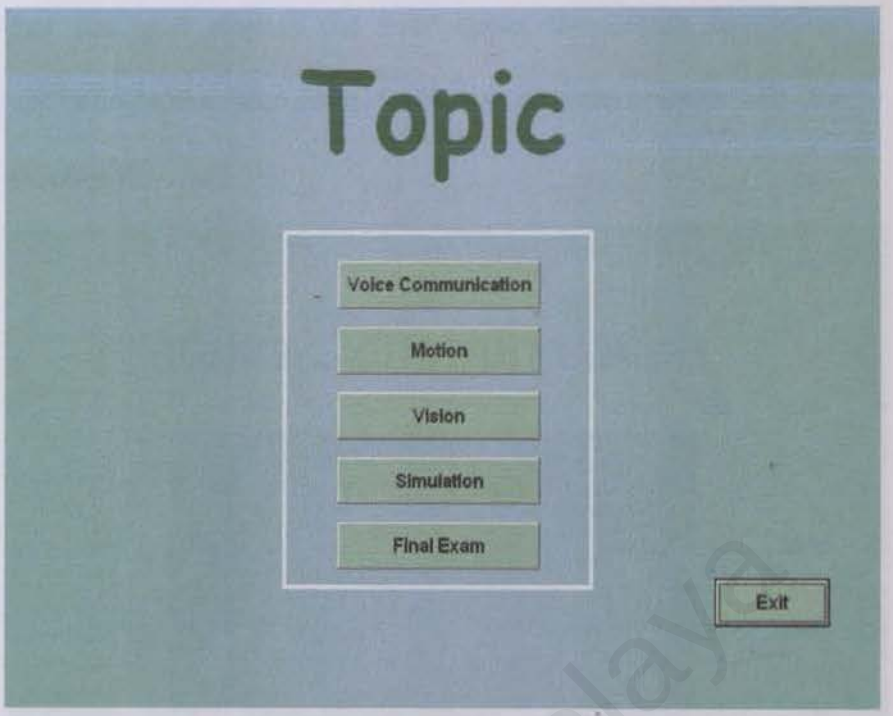


Figure 1.7 There are five optional buttons

6) For the next step if the user clicks the button for voice communication, the form for voice communication will be displayed.

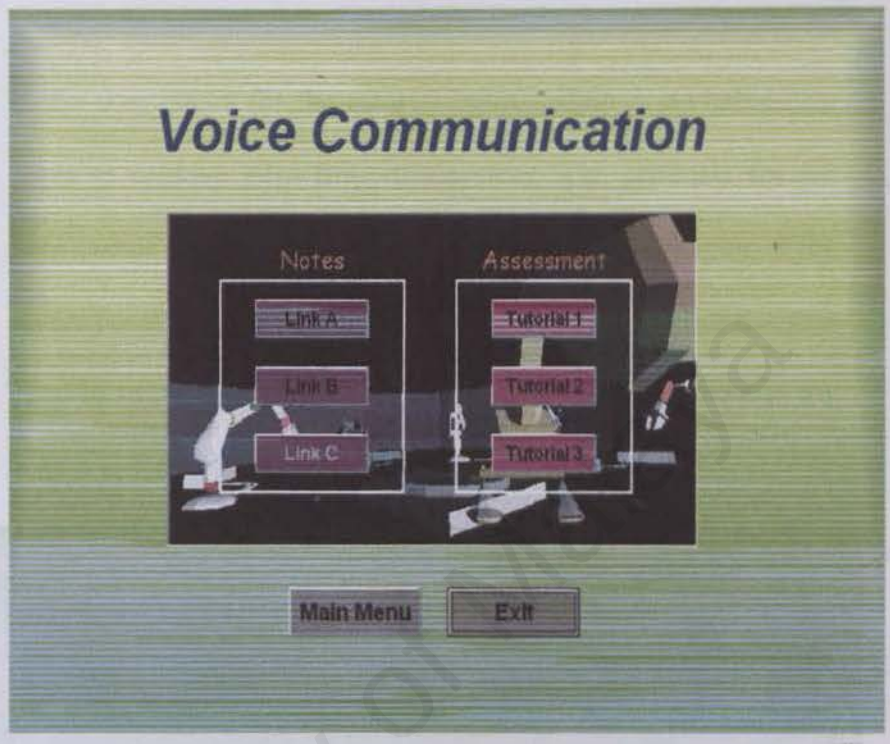


Figure 1.8 Student is optional either to take the notes that supply from the knowledge base or the assessment provided.

7) For the notes, if the type of user is intermediate Link A and Link B is enabled for those to retrieve the notes meanwhile the other links are disabled.

8) If the user clicks for the tutorial button, Help 2 is enabled for them to retrieve help from the system. The criterion for the help button is based on the result pretest for each chapter. If the result for the voice communication question is low so the first help is enabling, medium for the second help meanwhile the help 3 is for the highest result.

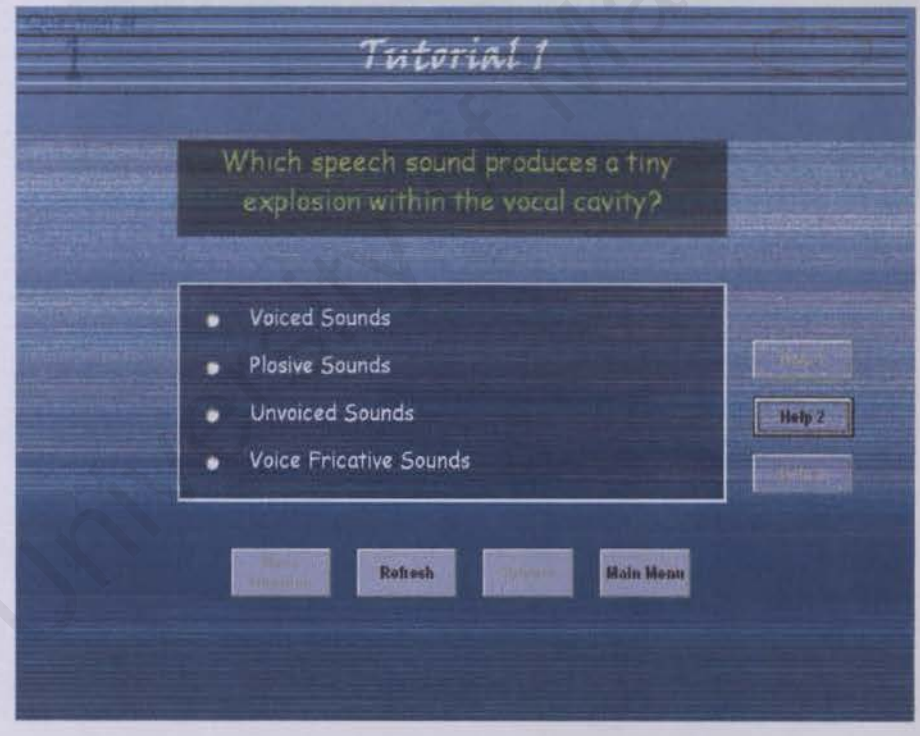


Figure 1.10 Help 2 is enabled for the user

9) User can also view for the simulation form which contains the rotation form and the obstacle avoidance to gain more knowledge in robotics.

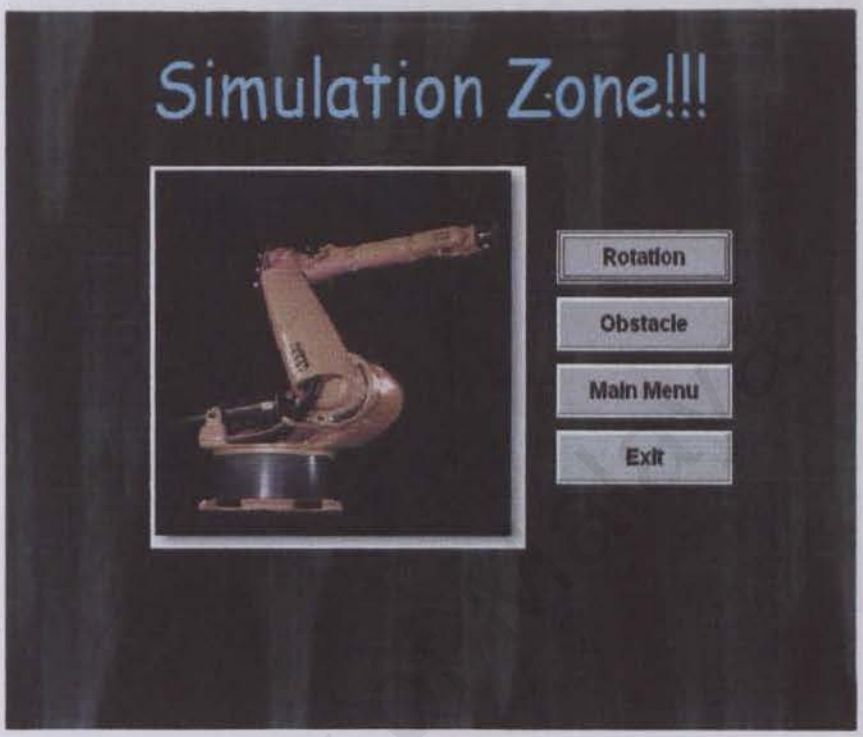


Figure 1.11 The simulation form

10) Finally, user can take the final exam to cater their knowledge in a robotic domain.

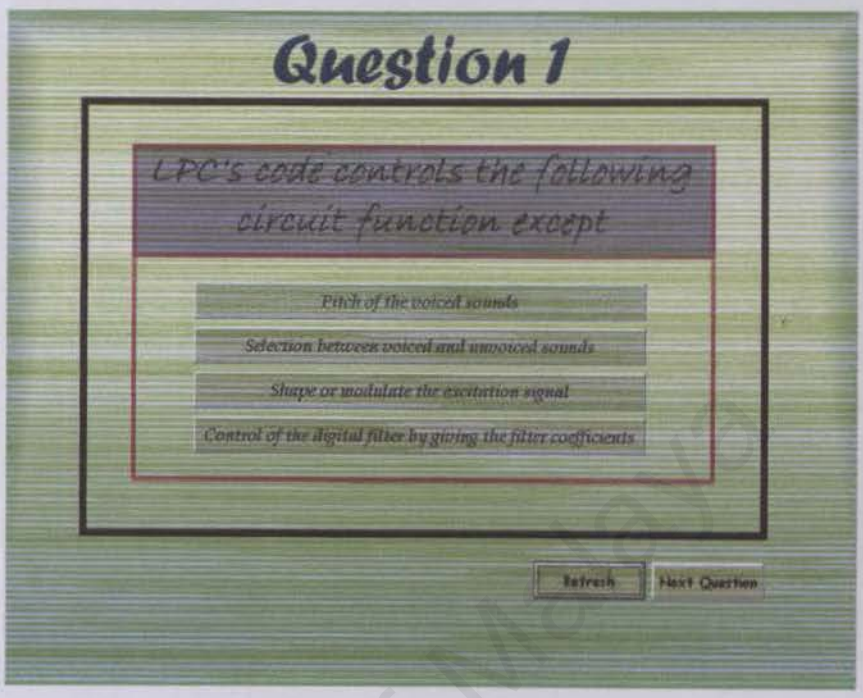


Figure 1.12 final exam forms

11) They also can view their report to analyze their score.

Figure 1.13 Report Final

12) Unloading form when user click the exit button

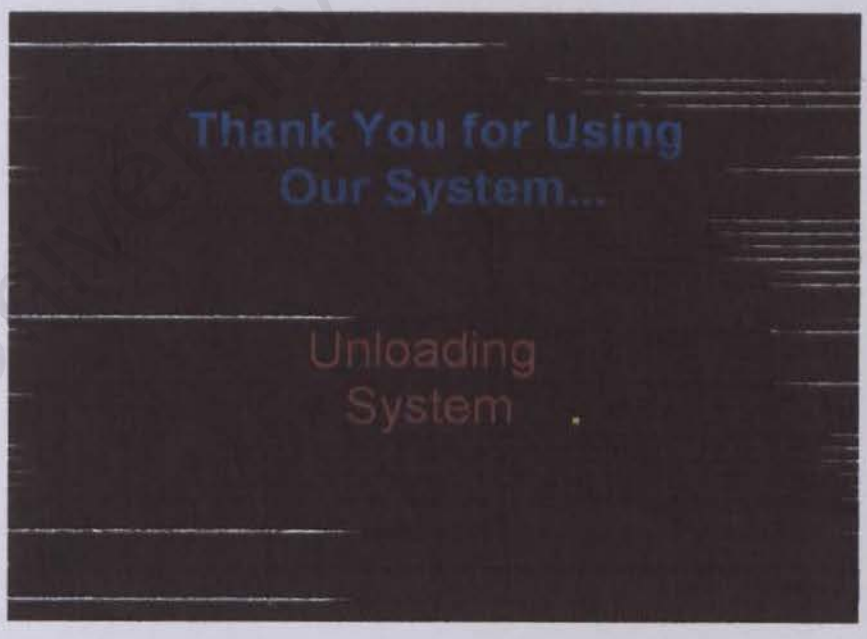


Figure 1.14 Unloading Form