**DOKUZ EYLÜL UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

# INTELLIGENT TUTORING SYSTEMS FOR EDUCATION

**by**

**KORHAN GÜNEL**

**July, 2006**

**İZMİR**

# INTELLIGENT TUTORING SYSTEMS FOR EDUCATION

**A Thesis Submitted to the**

**Graduate School of Natural and Applied Sciences of Dokuz Eylül University**

**In Partial Fulfillment of the Requirements for the Degree of Master of Science in**

**Computer Engineering, Computer Engineering Program**

**by**

**KORHAN GÜNEL**

**July, 2006**

**İZMİR**

## M. SC THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**INTELLIGENT TUTORING SYSTEMS FOR EDUCATION**" completed by **KORHAN GÜNEL** under supervision of **PROF. DR. TATYANA YAKHNO** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

. . . . . .

**PROF. DR. TATYANA YAKHNO**

Supervisor

. . . . . .

**Prof. Dr. Valery YAKHNO**

(Jury Member)

. . . . . .

**Yrd. Doç. Dr. Şen ÇAKIR**

(Jury Member)

---

**Prof. Dr. Cahit HELVACI**

Director

Graduate School of Natural and Applied Sciences

**ACKNOWLEDGEMENTS**

First and foremost, I would like to thank my supervisor Prof. Dr. Tatyana YAKHNO, for her guidance and encouragements to get me during my research. Also, I am grateful to Ph.D. student Rıfat AŞLIYAN and Ph.D. student Ferkan KAPLANSEREN for their valuable comments and suggestions on this study. We had have helpful discussions with them through the research.

Also, I thank my family for supporting me during this study.

KORHAN GÜNEL

**INTELLIGENT TUTORING SYSTEMS FOR EDUCATION**

## ABSTRACT

Intelligent Tutoring Systems (ITS) provide the benefits of one-on-one instruction using computers. ITSs can get up the performance of the students overcoming space, time, socioeconomic and environmental restrictions, according to lots of researchers. In this sense, the progress of researches to developed more effective programs to test and enhance the learning performance of students continues rapidly.

In this thesis, which aims to examine some characteristic properties of ITSs, an intelligent tutoring system for mathematics education at undergraduate and graduate level, has been developed. The developed system, which is called as MathITS, is based on conceptual map modeling. Hence, the thesis focuses on student modeling of system principally. Knowledge representation in the system is based on LaTeX notation to represent the mathematical symbols and notation easily. The Mathematica Kernel is used as an expert system in MathITS.

**Keywords:** Intelligent Tutoring Systems, Computer-Aided Instruction, Conceptual Map, Student Model, Mathematica, LaTeX.

**INTELLIGENT TUTORING SYSTEMS FOR EDUCATION**

## ÖZ

Akıllı öğretici sistemleri, bilgisayarları kullanarak bireysel öğrenme sürecinin avantajlarını sunarlar. Çoğu araştırmacıya göre akıllı öğretici sistemleri, zaman, mekan, sosyo-ekonomik ve çevresel sınırlamalardan bağımsız olarak öğrenci performanslarını arttırabilir. Bu bağlamda, öğrencilerin öğrenme yeteneklerini arttırabilecek yazılımların geliştirilebilmesi için araştırmalar hızla devam etmektedir.

Akıllı öğretici sistemlerinin karakteristik özelliklerini incelemeyi amaçlayan bu tezde, lisans ve yüksek lisans düzeyinde matematik eğitimi sağlayan bir akıllı öğretici sistem geliştirilmiştir. MathITS adı verilen bu sistem temel olarak kavramsal harita modeli üzerine kurulmuştur ve bundan dolayı tezde daha çok öğrenci modeli üzerine odaklanılmıştır. Matematiksel sembollerin ve notasyonun kolaylıkla ifade edilebilmesi amacıyla bilgi gösterimi için LaTeX simgelemi kullanılmıştır. Sistem içinde uzman sistem olarak Mathematica düşünülmüştür.

**Anahtar Sözcükler:** Akıllı Öğretici Sistemler, Bilgisayar Destekli Öğretim, Kavram Haritaları, Öğrenci Modeli, Mathematica, LaTeX.

**CONTENTS**

**Page**

# CHAPTER ONE
## INTRODUCTION

### 1.1 What is Intelligent Tutoring?

Intelligent Tutoring Systems are computer systems designed for support and improvement of learning and teaching process in the domain knowledge. They aim at adding computer aided instruction to traditional classroom teaching, one-to-one tutoring and individual learning. Computers have been used in education since sixties (Martin, 2001). Teachers and students use computers in all aspects of education such as researching, preparing study plan and organizing lecture notes, collecting grading information and doing homework. At the present time, it is hard to imagine a modern education without computers. The use of computers can be beneficial for teachers and learners.

Molnar (1997, p.10) points out "Research shows that educational technology, when properly applied, can provide an effective means for learning". With this viewpoint, this study aims to research the relationships between computers and education, the advantages of using computers in education and the pitfalls of using them. Hence, the thesis focuses on the Intelligent Tutoring Systems, referred to as programmed learning systems.

Murray (1998) describes the ITS such that "ITSs are computer-based instructional systems that have separate databases, or knowledge bases, for instructional content (specifying what to teach) and teaching strategies (specifying how to teach) and attempt to use inferences about a student's mastery of topics to dynamically adapt instruction." (Karlgren, 2005).

The definition given by Katie Hafner is that “Broadly defined, an intelligent tutoring system is educational software containing an artificial intelligence component. The software tracks students’ work, tailoring feedback and hints along the way. By collecting information on a particular student’s performance, the software can make inferences about strengths and weaknesses, and can suggest additional work.” (Hafner, 2000).

An ITS can be used to enable the students work independently, to improve their understanding of concepts within related domain, and to take progress of problem solving ability for each of them. On the other hand, an ITS can be able to assist not only their student users but also the teachers in developing and managing courses.

Intelligence involves mental capabilities such as the reasoning ability, planing, solving problems, thinking abstractly, comprehending ideas, and learning. Also it is related to creativity, personality, or character of the person according to psychology. On the other hand, mathematics is as a nightmare for lots of students. The students doubt their creativity, talent, and motivation when studying mathematics. This is unavoidable. In this sense, the tutoring systems must have the capability of real teachers, and it must act an human tutor in a class. ITS can rise up the effectiveness of teaching mathematics in a class. For that reason, the Intelligent Tutoring System is one of the subjects of AI. All the studies about intelligent tutors should be evolved depending on this thought.

In the 1970s, taking advantages of computer technology, researchers were looking for new educational paradigms, which provide the improvements in learning and problem solving. Thus, the combination of artificial intelligence, cognitive science and advanced technologies could be improved and Intelligent Computer-Aided Instruction (ICAI) came on the scene with this approach (Molnar, 1997).

Although Intelligent Tutoring Systems began with *Computer-Aided Instruction(CAI)*, they differs from them. Firstly, the interface, in CAIs, are always static

for each student and the information presented to each of students is exactly the same for all the time.

ITSs use the knowledge for pedagogical process so that the system try to determine what the student knows or does not know. Contrary to ITSs, CAIs have assumptions about what the student knows. Therefore, the same curriculum is presented to students in CAIs, even though the preceding knowledge is necessity for a student. This is the another difference between CAIs and ITSs.

The other difference between them is the feedback system. Some CAIs have the capability of asking questions to students. However the feedback system of them is restricted to indication of whether the student answer was correct or wrong, only. ITSs try to determine the students weaknesses on a topic using the domain and student model (see Chapter 2). They make the performance analysis and provide detailed feedback to each students. It is important that the student needs to know why the answer is wrong.

## 1.2 State of Arts

The theory of Intelligent Tutoring Systems began to evolve from Computer-Aided Instruction, in the early 1970s. In particular, with the viewpoint of mathematical education, specially undergraduate and graduate education, there is no significant theory of ITS. Although ITS theory began within the Artificial Intelligence (AI), the researches avoided to study in this area at the beginning. The main reasons of this avoiding are the lack of knowledge about mental models, language understanding and knowledge representation (Self, 1990). With the perspective of mathematical manner, the lack of symbolic computations can be added with this reasons.

Chronologically, some Intelligent Tutoring Systems, which have been developed so far, is summarized in the following.

*SCHOLAR*, Carbonell, Collins et al. (1969), was the earliest ITS and it was designed to teach South American geography. Its principal pedagogical strategy is the use of Socratic dialog in tutoring. It permits both students and tutor to ask questions. Then SCHOLAR poses questions to students and during the course of the dialog, it attempts to diagnose students' misconceptions.

*WHY*, Collins, Stevens et al. (1977), is an extension to research on the nature of tutorial dialogs done in SCHOLAR and uses Socratic dialog in an effort to teach learners the ability generalize from experience. It try to examine the effects of student's misconceptions.

*BUGGY*, Suppes (1981), was designed to help instructors to identify underlying misconceptions in student math problems. The tutor presents examples of incorrect math problems (buggy solutions). The systems then permits the instructor to generate their own problems for the system to solve in the incorrect manner. From this interaction, instructors attempt to understand the nature of a particular student misconception.

*SOPHIE*, (SOPHisticated Instructional Environment), Brown, Burton et al. (1982), is an electronic circuit debugging tutor. The system has knowledge of expert problem-solving strategies and a rule base to analyze and critique student attempts at solutions. Given a computer simulation of a faulty piece of equipment, students must locate faults by taking appropriate measurements.

*WEST*, Brown (1982), WEST uses coaching as its pedagogical approach, diagnosing student's misconceptions as they play a game entitled, "How the West Was Won," a computer-simulated board game for drill and practice in basic mathematics. WEST critiques students' decisions as they play, offering suggestions and comments to improve their performance.

*GUIDON*, Clancey (1983), is an expert instructional program built upon the earlier medical diagnostic program, MYCIN. GUIDON uses a mixed initiative dialog strategy in order to diagnose diseases.

*LISP Tutor*, Anderson et al. (1984), teaches the programming language LISP using a theory of learning called ACT*. ACT* is a general theory of cognition developed by John Anderson that focuses on memory processes. ACT* identifies procedural and declarative knowledge as part of its definition of expert problem-solving model. The system functions on the basis of a comparison of student performance to this model of expert performance using the ”model-tracing paradigm,” (see Section *1.5.1*) where the system responds to the students' divergences from the expert problem-solving path with hints directed toward guiding the student back onto the path. The LISP Tutor was the first in a series of tutors developed according to the principles of ACT*. Since the early 1980's, other ACT* tutors have been developed, including Pascal, Algebra, and Geometry.

*MAIS*, Tennyson, Park (1987), The Minnesota Adaptive Instruction System (MAIS) was developed according to the systems approach to instructional design. The MAIS adapts the rate at which students encounter topics using a Bayesian (predictive) statistical model of student progress through the curriculum. The difficulty level of lessons is adjusted according to a prediction of student performance based on a history of performance.

The following systems have been developed since 1990. *OGF*, Thepchai, S., Inaba, A., Mitsuru, I., Toyoda, J., Mizoguchi, T., Opportunistic Group Formation - researchers explored using intelligent agents as individual assistants in a collaborative learning environment. Agents assisted participants in identifying team goals as differentiated from personal goals by providing communication guidance to team process. Results not formally studied.

*MethodMan*, Crampe , Software Project Management. Explored user-controlled adaptation versus system-controlled adaptation. Experimentation with user-driven adaptivity proved that user-driven learning is limited by virtue of the learner's own lack of visibility into what their course of learning should be.

*ILEX*, Cox, R., O'Donnell, M., Oberlander, J., Intelligent Labeling Explorer - Intelligent hypertext online museum repository exploration. Researchers contrasted subject exploration patterns in two modalities: with intelligent system guidance, and without guidance. No significant learning differences noted in comparing these.

*ISIS*, Meyer, T., Miller, T., Steuck, K., Kretschmer, M., Instruction in Scientific Inquiry Skills - simulation-based cognitive tutoring system to teach junior and senior high school students scientific inquiry skills in biology and ecology. System used progressively difficult Skill Instructional Modules (SIMs). Two-years of quasi-experimental design study of effectiveness show that ISIS is more effective than traditional classroom instruction.

*PACT*, Aleven, V., Koedinger, K. R., Cross, K., Geometry instruction based on Andersonian ACT-R theory of instruction. Researchers investigated value of explanation requirement in student geometry problem-solving as compared with simple right-answer approach. Results showed that students providing reasons for their solutions showed a better understanding of geometry principles and were better at applying their knowledge when given new problems.

### 1.3 Architecture

A general architecture of an intelligent tutoring system is related to 4 different components. The major components of an ITS as simply depicted in Figure 1.1 are simply defined in the following.

- Domain and Expert Module: Simply, it represents the domain competence of the ITS, that is the ability to solve problems within the domain.
- Student Module: This module contains the diagnostic competence of the ITS and generates the student model with all information about the individual learner.
- Pedagogical Module: Also it is known as Tutor Module. It is responsible for the instructional competence and provides implementations of different tutoring strategies.
- Communication Module : It implements the human-computer-interface of the ITS.

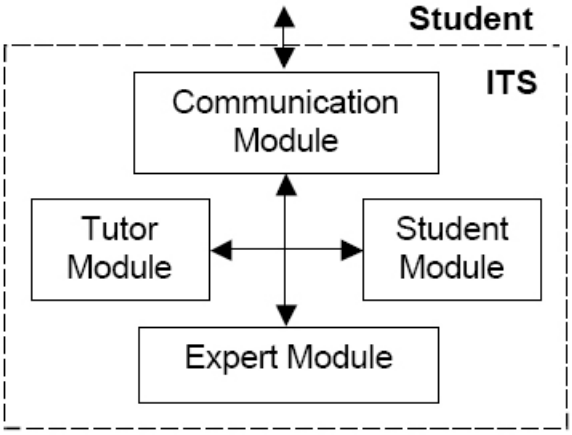


Figure 1.1 Classic ITS architecture (system border in dotted lines) (Devedzic & Harrer, 2005).

## 1.4 Domain Knowledge and Expert Module

Domain knowledge embedded in the system represents an experts knowledge and problem solving ability. An expert system aims to provide expert like solutions to problems in a specific domain. It often deal with uncertain and incomplete information and should be able to explain its decisions and underlying reasoning as human experts

are capable of doing. Figure 1.2 shown below illustrates a simple expert system architecture.

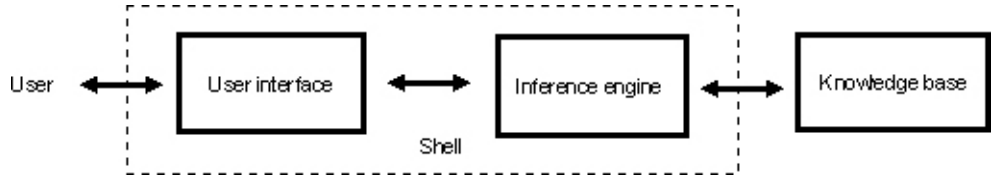


Figure 1.2 A simple architecture of an expert system

There are three modules within an expert system. These are the user interface which provides communication between the user and the system. The second module is the inference engine which is an interpreter for the knowledge base. It produces results and explanations for problems presented to it. The inference engine and the user interface are commonly viewed as a single component known as the expert system shell. The knowledge base contains the problem solving knowledge of a particular application. The knowledge enables the ITS to compare the learner's actions and selections with those of an expert in order to evaluate what the user does and doesn't know. The knowledge base itself is isolated from the expert system shell to allow reuse of the shell in other application domains.

## 1.5 Student Module

The Student Modeling Component is a critical component of an ITS, since the student has the central role in teaching process. Self (1990) addressed the critical necessities of student model.

The student model is simply about the theory of student behaviors. The student model evaluates each learner's performance to determine his or her knowledge, perceptual abilities, and reasoning skills. It provides the information such that what the student knows or does not know, any misconceptions, the student's degree of forget-

fulness and the facility of comprehension and accommodation of advices given by the system. Briefly, the student model determines model the student's mental state. Harp et al. (1995) pointed out that the student model must use the empirical student data, however most of student has handicapped in this sense and these student models do not sufficiently parameterized the student responses (see Harp, Samad, & Villano, 1995). There are several paradigm for student modeling, which are briefly summarized in following.

### *1.5.1 Model Tracing and Bug libraries*

This approach compares students action against an ideal procedure. Model tracing assumes that all of the student's significant mental states are available to the diagnostic program (Amižić, Stankov, & Rosić, 2002). The basic idea is to use an undetermined interpreter for modelling problem solving. At each step in problem solving, the under-determined interpreter may suggest a whole set of rules to be applied next, whereas a deterministic interpreter can suggest only a single rule. The diagnostic algorithm activates all these suggested rules, obtaining a set of possible next states. One of these states should correspond to the state generated by the student. The name model tracing comes from the fact that the diagnostic program merely traces the (undetermined) execution of the model and compares it to the student's activity. Input information for this algorithm is:

- expert knowledge and student's task
- set of productions
- student's answer that we want to trace

Output information is:

- True or false depending on whereas the student's answer has been traced.

- If the student's answer has been traced, output information is an array of chained production. If algorithm has not succeeded in finding an array of production that could generate student's answer, we say that the student's answer is not interpreted.

There are some productions that help us find wrong student's answers, that is, we make those answers understandable to the system. These "buggy" productions are used to make student's answer meaningful even though a student made a few wrong steps. The system generates feedback that informs student about his actions. There are two types of feedback:

- "buggy" feedback
- hints or help.

Every "buggy" rule generates message about mistake that has been made as a feedback. Hints or help are given on student's demand or when the system estimates that student needs them. Then the system generates an array of productions that represent a cognitive step a student should make to get himself to next step in problem solving process. That array of productions generates an array of hints that the system offers to student. Figure 1.3 shows how model-tracing paradigm works.

For instance, the production rules for question as seen in Figure 1.4 can be following.

**Two correct production rules:** IF goal is to find an angle in an isosceles triangle ABC and AC = AB and angle A is known THEN set the value of angle B to A.

IF goal is to find an angle in a triangle ABC and angles A and B are known, THEN set the value of C to 180-A-B.

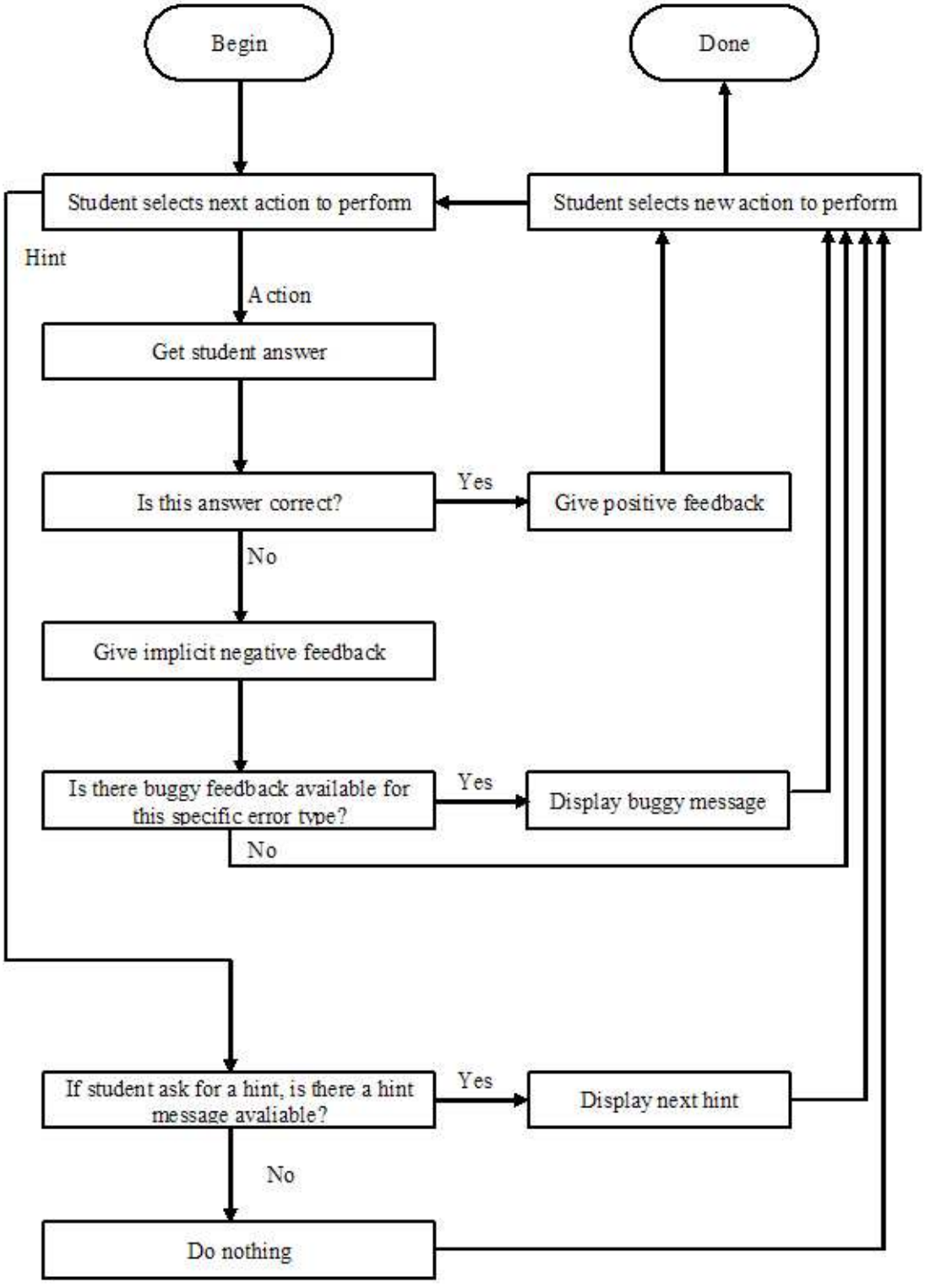


Figure 1.3 The general architecture for model tracing (Amižić, Stankov, & Rosić, 2002)

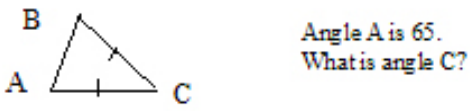


Figure 1.4 The question for computing the size of an angle

**Buggy production rule:** IF goal is to find an angle in an isosceles triangle ABC and angle A and C are at the bottom of the triangle and angle A is known THEN set the value of angle C to A.

Model tracing tries to dynamically simulate student's problem solving process and uses that simulation for interpretation of student's behaviour. This diagnostic technique bases itself on previously given catalogue of productions. Tracing starts after each student action and is being used for supervising student during his problem solving process. Model tracing is based on a idea of analyzing student cognitive process by reconstructing, step by step, process of making conclusions during a problem solving.

### *1.5.2 Overlay Model*

According to this model, the student's knowledge is seen as a subset of an expert's knowledge. Therefore, this model assumes that the student will not learn anything that the expert does not know. Figure 1.5 shows the general structure of this paradigm.

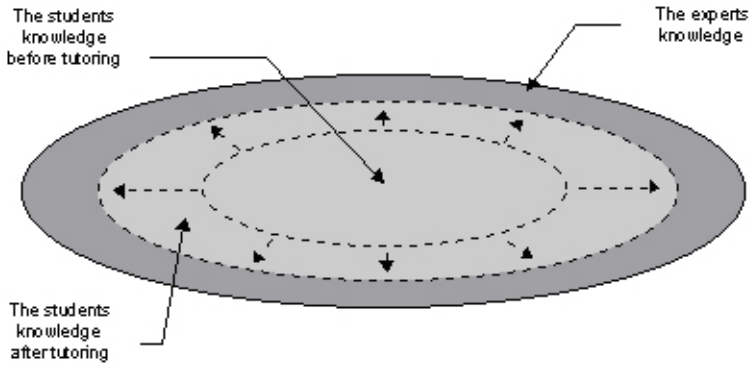


Figure 1.5 A representation of an overlay student model

The overlay model is more applicable when the subject matter to be learnt can be represented as a prerequisite hierarchy. In the prerequisite hierarchy, the subject has a number of concepts and for each concept there are a set of prerequisites. The overlay model indicates how far the student has progressed in acquiring the subject knowledge with respect to that of the expert. However, if the view of the tutor (expert) is different from that of the student then the overlay model may not be useful. GUIDON introduced by Clancey (1983) uses the Overlay Student Model.

### *1.5.3 Constraint-based Modelling*

This student modelling method describes only informative states pedagogically, rather than following the procedure the student used to arrive at their answer. The idea was introduced by Ohlsson in 1992 (Martin, 2001). The origin of the Constraint-based Modelling(CBM) is based on learning from performance errors. Originally, Ohlsson viewed CBM as an approach to developing short-term student models. However, Ohlsson did not give the details about how it can be implemented.

In CBM, domain knowledge represented by a set of constraints. A constraint is a pattern of form $< Cr, Cs >$. It means that if a solution matches the $Cr$ then it must also match the $Cs$, else something is wrong. For example, If the current problem is $\frac{a}{b} + \frac{c}{d}$, and the solution of the student is $\frac{(a+c)}{n}$, then it had better be the case that $n = b = d$.

The constraint set forms a detailed model of the domain, which partitions the solution space into one of three sets: for any given problem and student solution, a constraint may be relevant and satisfied, relevant but not satisfied, and not relevant (satisfied or not). The first set indicates a correct solution with respect to the current constraint; the second represents an incorrect solution; the third group tells us nothing about the solution at all. Therefore, to correct a wrong solution with respect to a given constraint, we have two choices: either satisfy the constraint, or render it irrelevant (Martin & Mitrovic, 2001). According to CBM, there is no need for an expert module.

### *1.5.4 Bayesian Net Modelling*

Bayesian networks, or belief networks, are graphical knowledge representations that handle uncertainty through a probabilistic formalism (Kai-min Chang, 2006).

A Bayesian network is a graphical knowledge representation in which each node of the network represents a random variable. The topology of the graph takes into account the dependencies between variables. More precisely, a Bayesian network (BN) is a directed cyclic graph whose nodes are in a one-to-one correspondence with n random variables' $x_1, ..., x_i$ such that where $P_0(x_j)$ is the set of parents of $x_i$ in the graph. In fact the structure of the graph is the translation of the relations of conditional dependencies between the variables. Once the network is defined, that is to say, the all its structure and probability tables are known, it is possible to proceed to different kinds of inferences. The evidence propagation consists in the calculation of the posterior probabilities knowing the values of some variables. It is an update of the probabilities, and the most used inference in BNs for student modeling. For that reason, the difficulties in applying Bayesian modeling are the high cost in knowledge acquisition and in the time to update the student model (Shang, Shi, & Chen, 2001).

The Andes physics tutoring system has a student modeler that uses Bayesian networks (Vanlehn, 2001).

## 1.6 Pedagogical Module

Pedagogical module make decisions related to the selection of appropriate learning units, learning methods, curriculum sequencing and questions (Jeremic & Devedzic, 2004). During such processes, the Pedagogical module communicates with the student model in order to get relevant information. According to ITS design, this module can generate the problems dynamically.

## 1.7 Organization of Thesis

This study is about the Artificial Intelligence in Education. In this work, which is aimed to collect the works about *Intelligent Tutoring Systems*, which are called ITSs shortly. Additionally, an application of ITSs on mathematics education has been developed to demonstration of conceptual map modelling (Hwang, 2003, Rehani & Sasikumar, 2003). The following steps were taken:

In Chapter I, briefly, ITSs are introduced and the application area of them have been given. The differences between Computer-Aided Instructions and ITSs were examined. Also, the available ITSs have been summarized.

Chapter II describes Intelligent Tutoring System, developed within this thesis, called by MathITS and the conceptual map modelling paradigm as a student modelling used in MathITS, were also introduced.

Chapter III deals with the manual of the MathITS application for mathematics education. It presents that how an expert prepares lecture notes and questions and how a student uses the MathITS.

Chapter IV summaries the concluding remarks on intelligent tutors and the future works about them.

# CHAPTER TWO
## MATHITS : AN INTELLIGENT TUTOR FOR MATHEMATICS EDUCATION

### 2.1 Architecture of MathITS

MathITS is an Intelligent Tutoring System for mathematics education which employs the conceptual map modeling technique (Hwang, 2003). It is a student-centrad system which supports interactive learning. In Figure 2.1, the general architecture of MathITS is given.

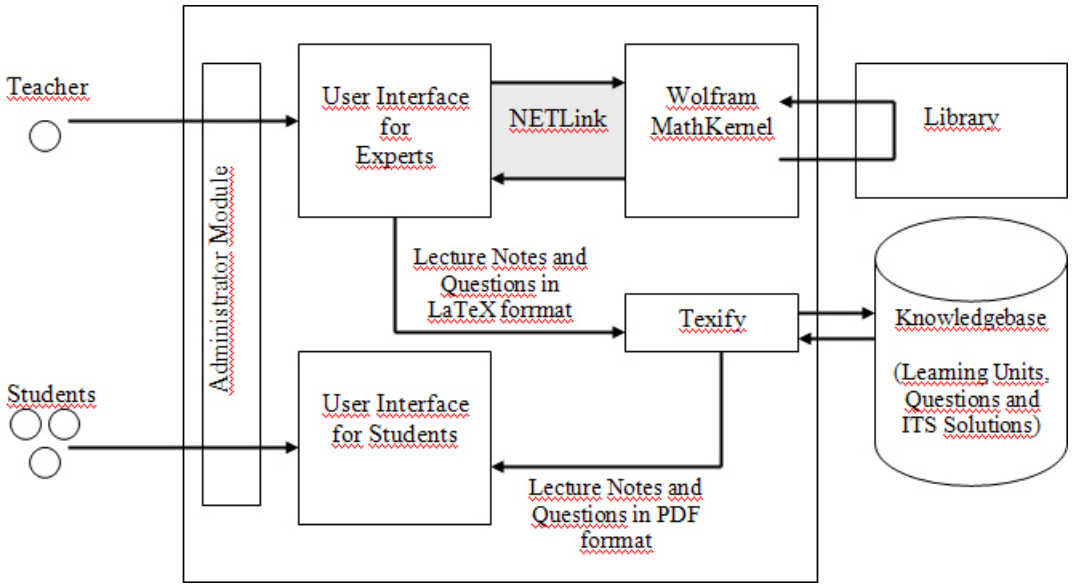


Figure 2.1 The general architecture for MathITS

MathITS has been developed as a Windows application and implemented with Visual Studio .NET v.2005. The system requirements are as given in Table A.1 in Appendix One.

As seen in Figure 2.1, the system has two types of users. Users login to the system through the administration module. Administration module checks and gets the user information from the database. If user exists, then the users are directed to an user interface according to their user types.

The expert users access the user interface shown in Figure 3.4. Expert users can write small lecture units, examples and questions. If an expert want to write question, he/she can use the Mathematica expression. Then the system executes the Mathematica Kernel by using *.NET/Link* component and evaluates the mathematical expression. The output is the MathKernel can be see in two different form:

- solution of the expression
- LaTeX notation of expression

## 2.2 Knowledge representation in MathITS

In addition, MathITS consists of a knowledge representation system. Because, knowledge representation systems have strengths unavailable to normal database systems. They allow a complex structural representation of the data. This allows inferencing and complex query evaluation to be performed. In the field of artificial intelligence, problem solving can be simplified by an appropriate choice of knowledge representation. Knowledge representation in MathITS is based on LaTeX. The general structure of the knowledgebase and the database are depicted in Figure B.1.

To illustration of the system, the domain was selected as small as possible in the thesis. The only one lecture unit has been added into the knowledgebase is ”Calculating Area with Parametric Equations”, which its knowledge representation is as given Appendix Two. The knowledgebase stores 24 different concepts as seen in Appendix Four and a sample test, involves totally 11 questions, is shown in Appendix Three.

The original TeX(Tau epsilon Chi) system built by Donald Knuth. TeX is a computer language designed for use in typesetting; in particular, for typesetting math and other technical material (The TeX Users Group TUG, 2006). It converts marked up documents (TeX files) into a formatted DVI (Device Independent format) file for viewing or conversion to another format. Although TeX is a relatively low-level language, it is expandable and common TeX actions can be combined into macros. Thus, Tex itself can be quite a challenge to use. The most successful set of such macros is called LaTeX, which designed by Leslie Lamport (The TeX Users Group TUG, 2006). Conversely MathITS uses LaTeX. In MathITS, all the knowledge such as small lecture units, questions, answers and hints is stored in LaTeX format, and the knowledge is converted to PDF documents when necessary.

Originally LaTeX is not a word processor! MathITS have an user interface to connect LaTeX shell. This part of system acts as a graphical front end for LaTeX. This is basically a text editor that communicates with the LaTeX program. It may also highlight TeX keywords and provide other useful functions.

Also the user interface consists of an ActiveX component, which is used for document viewer. The all the knowledge is presented to students with a Portable Document Format(PDF) viewer.

The Mathematica kernel is basically an interpreter for the Mathematica programming language. This language combines features from procedural, functional, and rule-based programming.

The MathITS has an interface to connect the Mathematica Kernel. This interface supports that an expert can enter Mathematica commands and receives the results in text-based, graphics or LaTeX notation of the command. Figure 2.2 shows this paradigm.

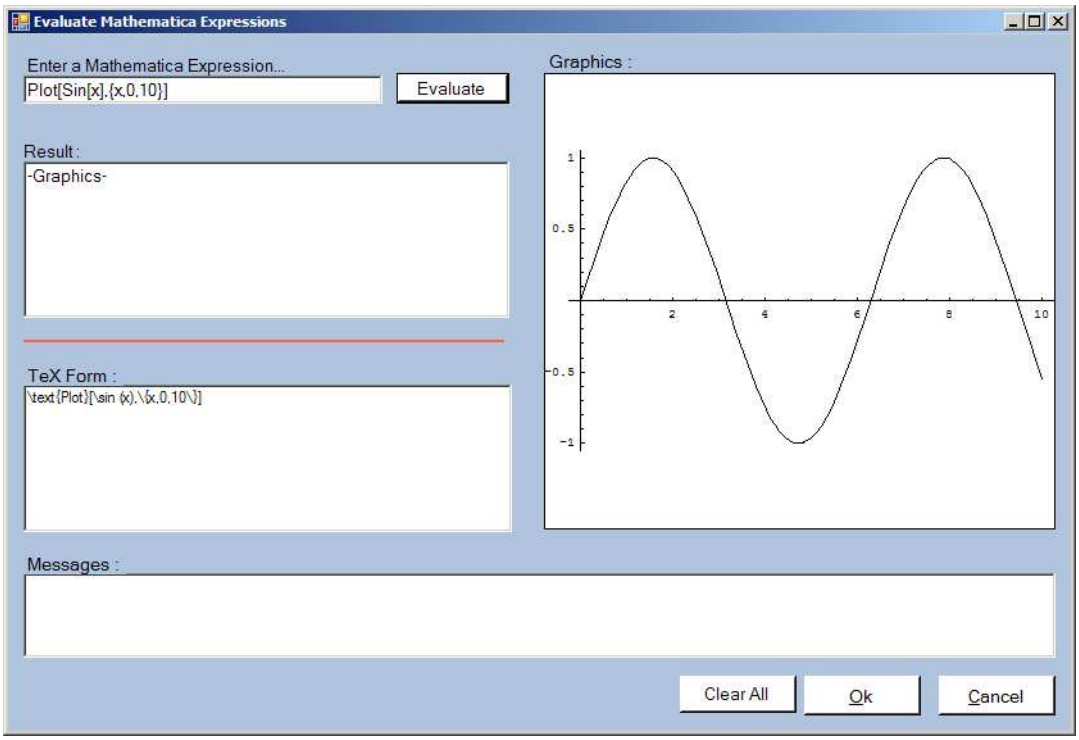


Figure 2.2 Experts can evaluate the Mathematica expression.

The Mathematica library incorporates the system's "expert knowledge"; it consists of a large number of mathematical algorithms written in the Mathematica language and interpreted by the Mathematica kernel. The communication between the Math-ITS's user interface and Mathematica Kernel is supported by *.NET/Link* component. *.NET/Link* integrates Mathematica and Microsoft's .NET platform. *.NET/Link* lets you call .NET from Mathematica in a completely transparent way, and allows you to use and control the Mathematica kernel from a .NET program. *.NET/Link* uses *MathLink*, the protocol, defined by Wolfram Research, for sending data and commands between programs. Many of the concepts and techniques in *.NET/Link* programming are the same as those for programming with the *MathLink* C-language API (Wolfram, 2002).

## 2.3 Student Modelling in MathITS

In tutoring systems, students can learn new concepts and the relationships between the previously learned and the new concepts. This knowledge is represented as a

conceptual map in MathITS. Hwang (2003) introduced the “concept effect graphs” where the subject materials can be viewed as a tree diagram comprising chapters, sections, sub-sections and key concepts to be learned. Figure 2.3 simply illustrates the tree structure diagram for the number concept. However, the concept maps can be more complex structures, so it can not be represented as tree structure always. Originally, they should be thought as directed graphs.

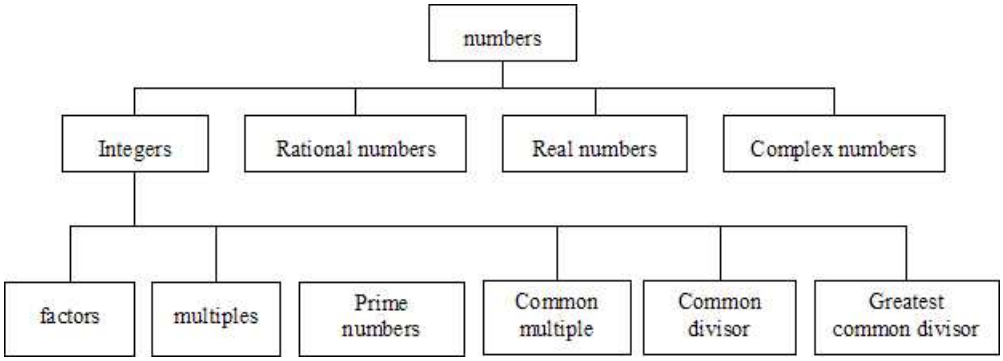


Figure 2.3 Tree structure diagram for the number concept.

The conceptual map approach offers an overall cognition of the subject contents. With this approach,diagnosis process can be made easily. If a student fails to learn the concept “common divisor”, it is possible that the student did not learn the concept “factors”. Therefore the system suggest that the student must study the factor concept again. For this reason, the ITS must contain the relationships between concepts. To do this,a conceptual map-based notation is proposed. Suppose that $C_i$ and $C_j$ are two concepts and if the concept $C_i$ is prerequisite for the concept $C_j$ then a concept effect relationship $C_i \rightarrow C_j$ exists. Of course, a single concept may have multiple prerequisite concepts, and it can be a prerequisite concept for multiple concepts. Figure 2.4 illustrates the concept effect graph for the number concept.

To construct the concept effect graph in MathITS, the Concept Effect Table(CET) is used, which represents the relationships between the concepts to be learned. Table 2.1

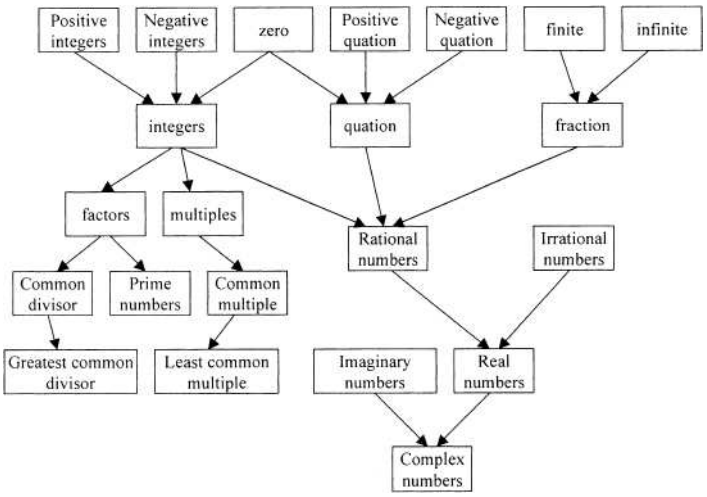


Figure 2.4 The concept effect graph for the number concept (Hwang, 2003).

demonstrates how the CET is constructed. As seen in Table 2.1, if $CET(C_i, C_j) = 1$ then $C_i$ is the one of prerequisite concept for concept $C_j$ and $NP_j$ represents the total number of the prerequisite concepts for $C_j$.

Table 2.1 An Example of the Concept Effect Table

| | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_j$ | $\cdots$ | $C_n$ |
|---|---|---|---|---|---|---|---|
| $C_1$ | 0 | 0 | 1 | $\cdots$ | 0 | $\cdots$ | 0 |
| $C_2$ | 0 | 1 | 1 | $\cdots$ | 0 | $\cdots$ | 1 |
| $C_3$ | 0 | 0 | 0 | $\cdots$ | 1 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | | | $\vdots$ |
| $C_i$ | 0 | 0 | 0 | $\cdots$ | 1 | $\cdots$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | | | $\vdots$ |
| $C_n$ | 0 | 0 | 0 | $\cdots$ | 0 | $\cdots$ | 0 |
| $NP_j$ | 0 | 1 | 2 | $\cdots$ | 2 | $\cdots$ | 3 |

Table 2.2 demonstrates the difficulty rates for each questions. The question difficulty rates ($QDRT$) are calculated as the ratio of the total number of related concepts with a question ($Nc$) to total number of concepts ($n$). The mark of the question $Q_i$ is calculated as $10 \times \frac{Nc_i}{n}$. According to Table 2.2, the test contains 10 questions and $n$ different concepts. Thus, the QDRT value of the question $Q_5$ is $QDRT(5) = \frac{3}{n}$ and the mark of this question is $10 \times \frac{3}{n}$ points. With this calculation, the difficulty of a

question has a value in the range of interval $(0, 10]$. Suppose that a test contains $m$ different question and $n$ different concepts. Then total mark for this test is calculated in Formula 2.1.

$$TotalMark = 10 \times \frac{\sum_{i=1}^{m} Nc_i}{n} \tag{2.1}$$

Table 2.2 The Question Difficulty Rate Table (QDRT)

| | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_j$ | $\cdots$ | $C_n$ | **Nc** |
|---|---|---|---|---|---|---|---|---|
| $Q_1$ | 1 | 1 | 0 | $\cdots$ | 0 | $\cdots$ | 0 | 6 |
| $Q_2$ | 0 | 1 | 1 | $\cdots$ | 0 | $\cdots$ | 0 | 5 |
| $Q_3$ | 1 | 0 | 1 | $\cdots$ | 1 | $\cdots$ | 0 | 4 |
| $Q_4$ | 0 | 0 | 0 | $\cdots$ | 1 | $\cdots$ | 1 | 4 |
| $Q_5$ | 1 | 0 | 0 | $\cdots$ | 0 | $\cdots$ | 0 | 3 |
| $Q_6$ | 1 | 1 | 0 | $\cdots$ | 1 | $\cdots$ | 0 | 2 |
| $Q_7$ | 0 | 1 | 1 | $\cdots$ | 0 | $\cdots$ | 0 | 2 |
| $Q_8$ | 0 | 0 | 1 | $\cdots$ | 1 | $\cdots$ | 0 | 1 |
| $Q_9$ | 1 | 1 | 0 | $\cdots$ | 1 | $\cdots$ | 1 | 1 |
| $Q_{10}$ | 1 | 1 | 0 | $\cdots$ | 0 | $\cdots$ | 1 | 1 |

Table 2.3 The Test Item Relationship Table (TIRT)

| | $C_1$ | $C_2$ | $C_3$ | $\cdots$ | $C_j$ | $\cdots$ | $C_n$ |
|---|---|---|---|---|---|---|---|
| $Q_1$ | 5 | 1 | 0 | $\cdots$ | 0 | $\cdots$ | 0 |
| $Q_2$ | 0 | 4 | 2 | $\cdots$ | 0 | $\cdots$ | 0 |
| $Q_3$ | 2 | 0 | 3 | $\cdots$ | 1 | $\cdots$ | 0 |
| $Q_4$ | 0 | 0 | 0 | $\cdots$ | 3 | $\cdots$ | 5 |
| $Q_5$ | 1 | 0 | 0 | $\cdots$ | 0 | $\cdots$ | 0 |
| $Q_6$ | 4 | 1 | 0 | $\cdots$ | 2 | $\cdots$ | 0 |
| $Q_7$ | 0 | 5 | 2 | $\cdots$ | 0 | $\cdots$ | 0 |
| $Q_8$ | 0 | 0 | 5 | $\cdots$ | 2 | $\cdots$ | 0 |
| $Q_9$ | 3 | 2 | 0 | $\cdots$ | 4 | $\cdots$ | 4 |
| $Q_{10}$ | 1 | 1 | 0 | $\cdots$ | 0 | $\cdots$ | 1 |
| $Sum(C_j)$ | 16 | 14 | 12 | $\cdots$ | 12 | $\cdots$ | 10 |

Hwang (2003) also proposed the Test Item Relationship Table(TIRT) to calculate the total strength of a concept as shown in Table 2.3. If a test sheet contains 10 questions on a learning unit, $TIRT(Q_i, C_j)$ represents the intensity of the relationship

between the question $Q_i$ and the concept $C_j$. The intensity value is specified between 0 and 5 in MathITS. While 0 value indicates no relationship and 5 intensity value represents the most strong relationship between the question and the concept. In Table 2.3, $Sum(C_j)$ value presents the total strength of concept $C_j$.

Briefly, the TIRT table is used to calculate the probability of failing to answer for a student. If a student fails to answer only the question $Q_4$, the student will fail to answer 25% of questions related to the concept $C_j$ and 50% of questions related to the concept $C_n$. This probability is calculated as the ratio of $TIRT(Q_4, C_j)$ to $Sum(C_j)$ and the ratio of $TIRT(Q_4, C_n)$ to $Sum(C_n)$. If the student fails to answer more than one questions, then the Formula 2.2 can be used to calculate failing rates for each concept. Assume that vector $\mathbf{A}$ represents the students answers for each of the questions and the test has $m$ questions. If $i^{th}$ index of vector $\mathbf{A}$ is 1 then the student answer the question $Q_i$ correctly, otherwise $\mathbf{A}_i = 0$.

$$P(C_j) = \frac{\sum_{i=1}^{m}(1 - \mathbf{A}_i) * TIRT(Q_i, C_j)}{\sum_{i=1}^{m} TIRT(Q_i, C_j)}, \qquad j = 1, 2, \ldots, n \tag{2.2}$$

In Formula 2.2, $P(C_j)$ represents the probability of failing to answer for the concept $C_j$. Let vector $(A)$ be $\mathbf{A} = (1, 1, 1, 0, 1, 0, 1, 1, 0, 0)$, for $m = 10$ and $TIRT$ is as given Table 2.3. Then the student will fail to answer 50% of questions related to the concept $C_1$, 28.5% of questions related to the concept $C_2$, 58.33% of questions related to the concept $C_j$. According to the demonstration, it seen that the student does not understand the concept $C_n$, because the $P(C_n) = 1$. For that reason, the system advices to the student for studying the concepts related to the concept $C_n$. As seen in Table 2.1, the system can easily determine the related concepts using Concept Effect Table(CET). It suggest that the student should study the concepts $C_2, C_3$ and $C_i$.

Table 2.4 The Student Learning Status related to a concept.

| $P(C_j)$ | Learning Status |
|---|---|
| $> 0.75$ and $<= 1$ | Very poorly learned |
| $> 0.50$ and $<= 0.75$ | Less poorly learned |
| $> 0.25$ and $<= 50$ | Learned |
| $>= 0$ and $<= 0.25$ | Very well learned |

According the ratio of incorrect answer provided by a student related to concept $C_j$, the student's learning status of this concept is specified as in Table 2.4. These values are used in MathITS, however they are not certain standard. The range is only determined intuitively in this study. Our example shows that while the student has learned the concept $C_1$ and $C_2$, the concept $C_j$ is less poorly learned and the concept $C_n$ is very poorly learned by the student.

The CET, QDRT and TIRT tables in knowledgebase of MathITS have been constructed, so far, in Appendix Four.

# CHAPTER THREE
## USER GUIDE OF THE MATHITS

When a user runs the application, he/she will be directed to a login form after a splash screen depicted as in Figure 3.1. After the user writes her/his username and password, the Administration Module of MathITS checks the user name and password, as seen in Figure 3.2. If the user gets an approval from Administration Module, the next task of this module is determining the type of user. MathITS has three different user type, which are system administrator, student and teacher. Each of these users have different roles. In the following, the scenario is introduced for each type of user.


Figure 3.1 The Slapsh Screen

### 3.1 User Guide for Admin

The system administrator is the principal user for the system. The admin manages the login information and accessing grants of the users. He/she can create an account

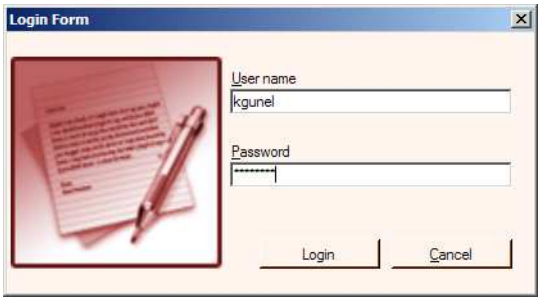


Figure 3.2 Login Form of the system

for a new user and determine the user type. The authority of removing any user from system is reserved by an admin. Figure 3.3 shows the user interface, when an admin logs into the system. The admin can view all active users in a table on this interface.

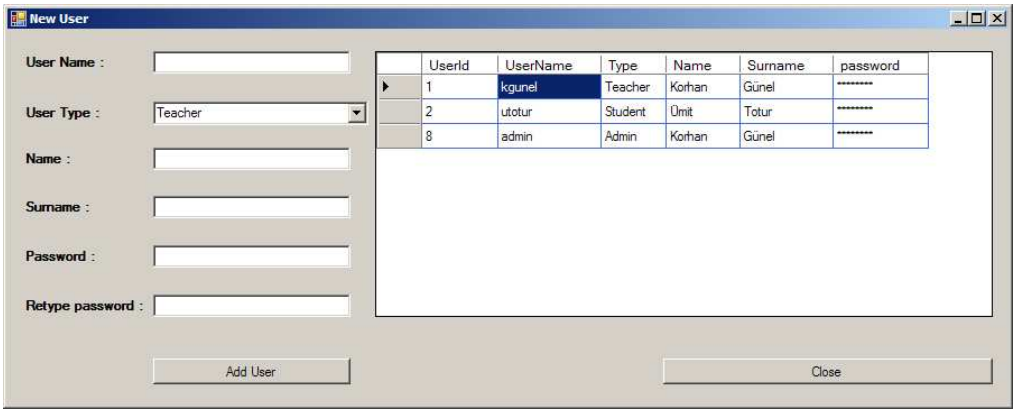


Figure 3.3 User Interface for an admin

## 3.2 User Guide for Teachers

The teachers has an essential role in MathITS and have lots of task to do. Probably, the most complex scenario within MathITS belongs to teachers. With this scenario, a teacher is able to

- create a new book,
- add a chapter into a book,

- create concepts, which will be learnt to a student in a chapter,
- associate a concept with others. Thus, he/she can generate a concept map for a chapter.
- add a section into a chapter of a book,
- create a lesson in a section,
- create a test question associated with a lesson
- use a Mathematica expression to solve and generate the choices of the question,
- associate a question with a lesson,
- connect a question with all the concepts in a chapter with specifying the weights of relation between the question and the concepts. Thus he/she specify the difficulty level of a question.

When a teacher logs into the system, the Administration Module calls the "*Expert Module Form*" as shown in Figure 3.4.

The user interface for a teacher acts as a highlighted LaTeX editor. However, the users need not to know the LaTeX, because the user interface is so friendly. When a user wants to enter a LaTeX command into Text box in the form, it is enough to click related button on the Toolbars or to select the related command from the *Insert* or *Functions* menus as shown in Figure 3.5 and Figure 3.6.

*File Menu* contains the basic file options for creating, opening, saving and printig lecture notes and questions in LaTeX form as draft. In this menu, *New Lecture Unit* command creates a new TeX file as a lecture unit. If user selects this command, the following text is written into the editor.

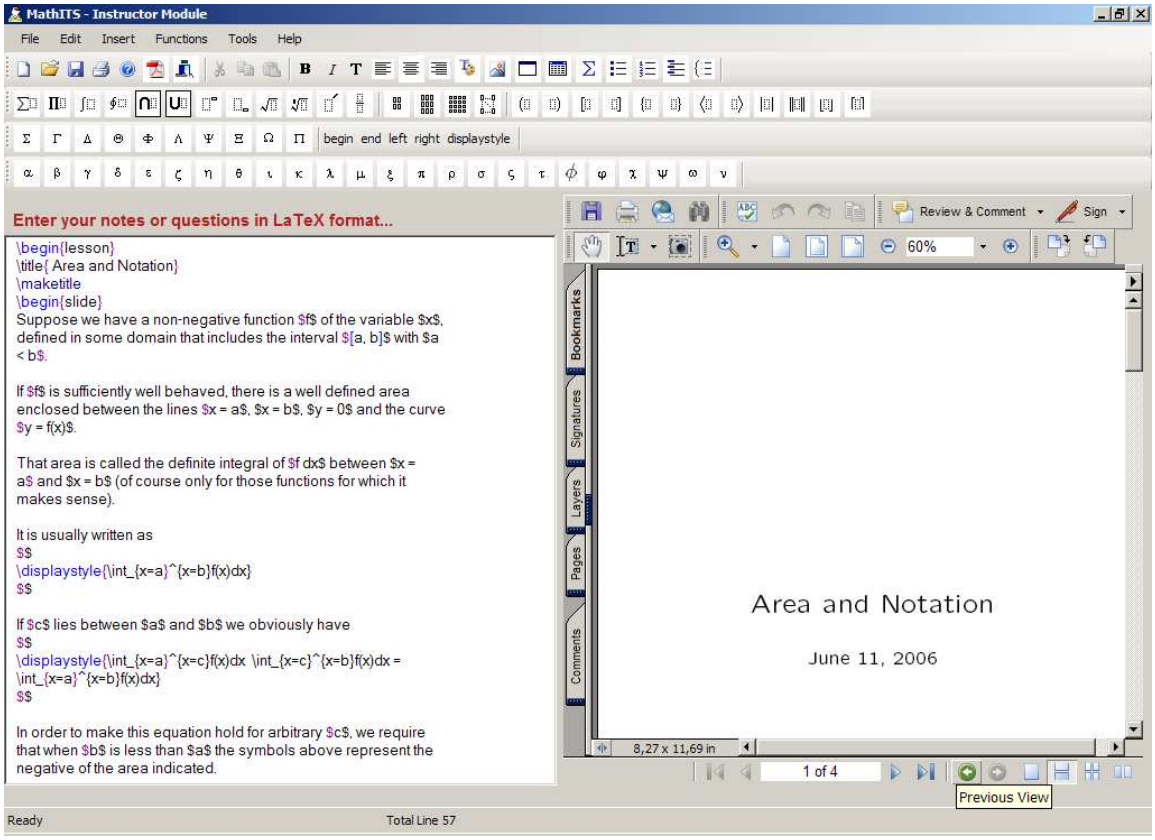


Figure 3.4 The user interface for a teacher

```
\begin{lecture}
    \title{ }
    \author{ }
    \date{ }
    \maketitle
    \begin{slide}
    \end{slide}
\end{lecture}
```

Similarly, *New Question* command creates a new TeX file as a lecture unit. If user selects this command, the following text is written into the editor.

```
\begin{question}
\begin{description}
```

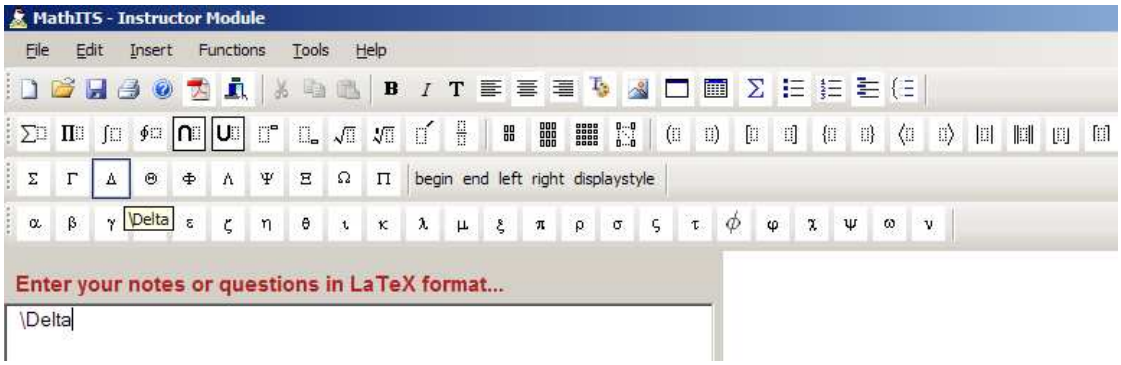


Figure 3.5 Toolbars in MathITS

```
    \item[\textbf{A)}]
    item[\textbf{B)}]
    item[\textbf{C)}]
    item[\textbf{D)}]
    item[\textbf{E)}]
\end{description}
\end{question}
\begin{hint}
\end{hint}
\begin{answer}
\end{answer}
\begin{solution}
\end{solution}
```

This commands are not standard LaTeX commands, and only defined in MathITS. All the questions in MathITS are defined as multiple choices questions. *Open* command opens an existing TeX file. *Save* command saves the contents of editor in a TeX file as a draft. *Save As* command saves the the contents of editor under a different name. *Print* command outputs the lecture units and questions in LaTeX notation to a printer, directly. *Print Preview* command preview users printed content on the computer screen. *Exit* command ends the execution of the program.

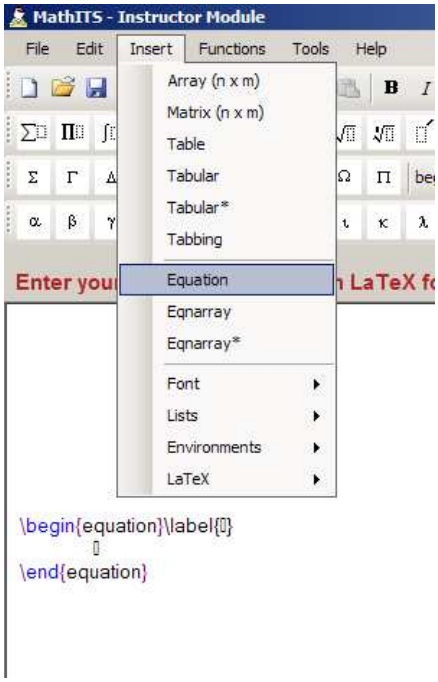


Figure 3.6 Menus in MathITS

*Edit Menu* allows users to manipulate text in the editor area. In this menu, *Undo* command reverses users most recent editing action. *Redo* command re-applies the editing action that has most recently been reversed by the *Undo* command. *Cut* command removes the selection and places it on the clipboard. *Copy* command places a copy of the selection on the clipboard. *Paste* command places the text or object on the clipboard at the current cursor location in the editor. *Select All* command selects all text or objects in the editor.

*Insert Menu* provides users to add standard LaTeX commands, such as equations, arrays, matrix, tables, into the editor. For Example if the user selects the *Table* command from the *Insert Menu*, then the following text is inserted in to the editor.

```
\begin{table}
    \centering
    \caption{*}\label{*}
\end{table}
```

*Functions Menu* includes some of standard mathematical functions such as $\sin, \cos, \max, \min, \exp, \log$ etc. When user selects one of the commands from this menu, the LaTeX notation is inserted into the editor.

*Tools Menu* helps experts to write and compute Mathematica expressions. When user selects *Create Math Expression* commands from this menu, *Evaluate Mathematica Expression Form*, shown in Figure 2.2, is opened. With this form, experts can easily enter a Mathematica Expression into the textbox under the label *Enter a Mathematica Expression....* After entering an expression, user clicks the *Evaluate* button and the Mathematica Kernel solves the problem. The Mathematica Kernel is basically an interpreter for the Mathematica programming language. This language combines features from procedural, functional, and rule-based programming (Wolfram, 2002). The kernel uses the Mathematica Library, which incorporates the system's "expert knowledge". It consists of a large number of mathematical algorithms written in the Mathematica language.

For instance, when an expert wants to plot graphics of $\sin$ function, he/she writes the command "$\mathrm{Plot[Sin[x], x, 0, 10]}$" into the textbox and then clicks the *Evaluate* button. Then the Mathematica Kernel outputs the type of the result in *Result box* as "$-\mathrm{Graphics}-$", the LaTeX notation into the *TeX box* as "$\mathrm{Plot[\sin(x), \{x, 0, 10\}]}$" and plots the graphics in *Graphics box*. Also, if an expert want to solve $\int_0^{\pi} \sin x dx$, he/she writes $\mathrm{Integrate[Sin[x], x, 0, Pi]}$ as a Mathematica expression, then the kernel output the results as $2$ and the LaTeX notation of the problem as

```
\int_0^{\pi } \sin (x) \
dx
```

kernel outputs a warning message into the *Messages box* as depicted in Figure 3.7. The warning messages are retrieved from the Mathematica Library.

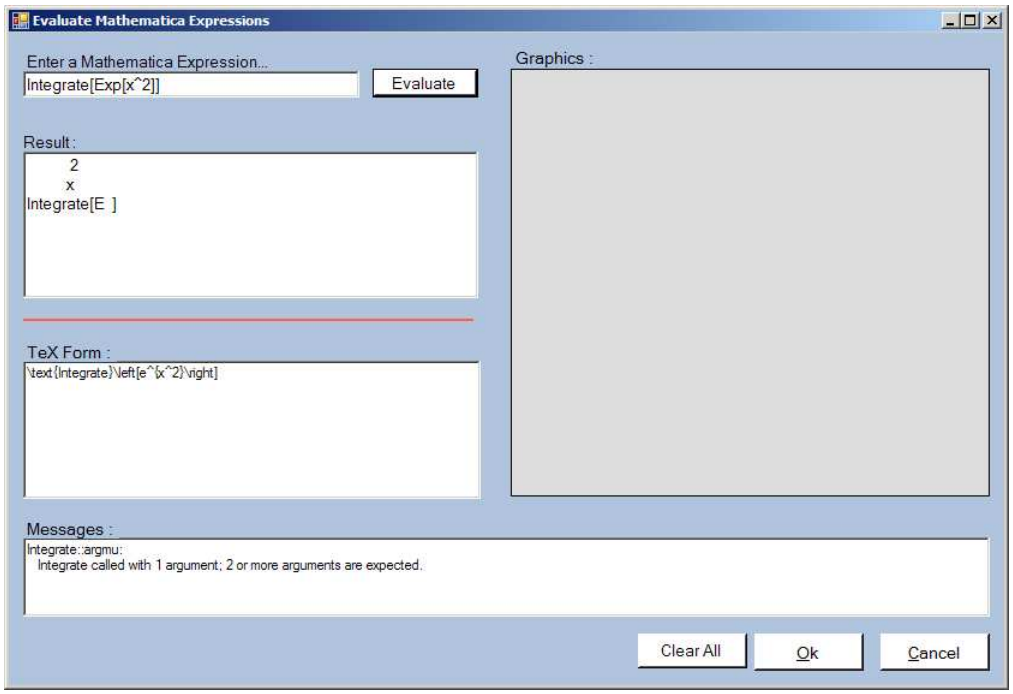


Figure 3.7 Warning Messages for Mathematica Expressions

Most of the commands defined in menus are accessible as one click button in the Toolbars, and several of them have already assigned to keyboard shortcuts.

Although it is not a necessity to know the all the LaTeX commands by the user, the system has an help file about LaTeX commands. To view the help file, the user must select the *LaTeX2e Reference* command from the *Help* menu. The help file contains all standard LaTeX commands categorically, as seen in Figure 3.8. Thus, the expert can easily add the small lecture units and questions to his/her lecture notes in TeX format in Figure 2.2.

Now we can start our scenario. When a teacher logs into the system at first time, he/she must do the following step by step. He must create a new book, by selecting the "*Add New Book*" command from the "*Tools Menu*". Then the following dialog depicted, as Figure 3.9, will be visible in the center of screen. As seen, the system

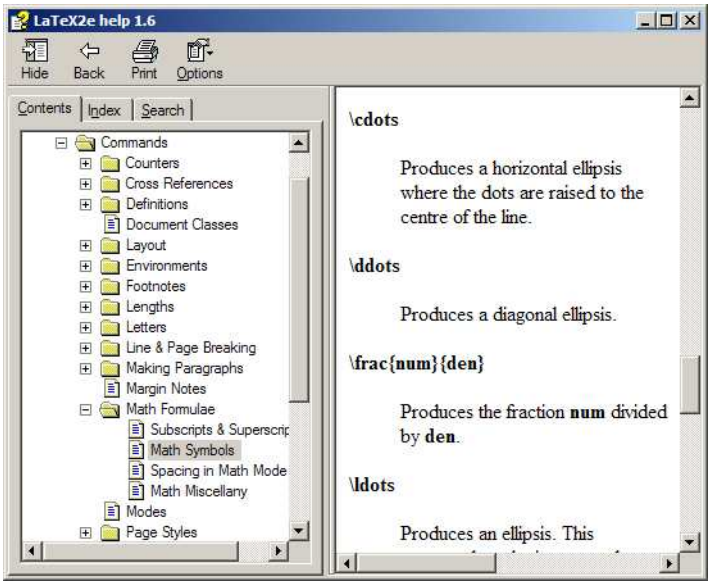


Figure 3.8 The Help File for LaTeX Reference

lists see all the available books. Also the user can delete any of these book listed in table.

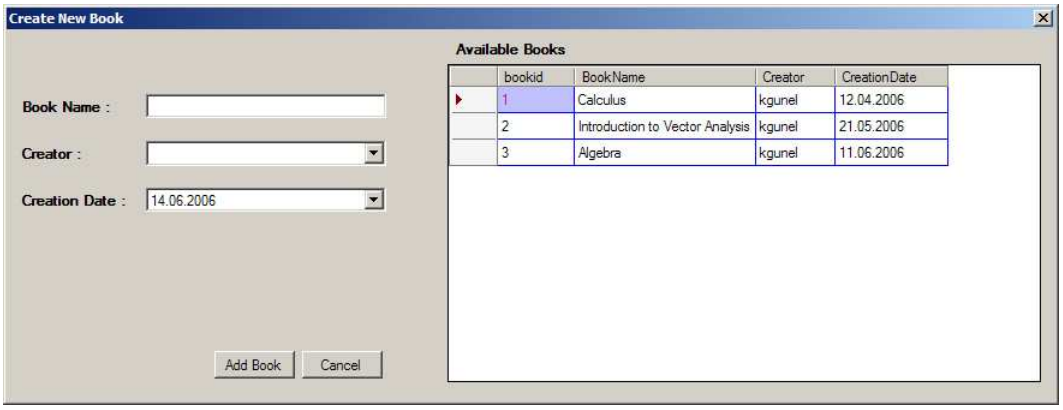


Figure 3.9 Create New Book Dialog

The second step is to create a chapter within a book. For this aim, the user must select the "*Add New Chapter*" command from the "*Tools Menu*". After the command is selected, the dialog, which is shown in Figure 3.10, can be able to viewed by the user. When the user selects a chapter from the combobox on the dialog screen, all the available chapters within the selected book are listed. The next step is the specifying

all the concepts which will be used during the chapters. Figure 3.11 illustrates how a concept can be added by a teacher. When a user select a book from "*Book Name*" combobox, all the chapters are listed, and in an analogous manner, when a chapter is selected all the available concepts are shown. Teacher must determine the necessity of a concept by thinking the chapter objectives.

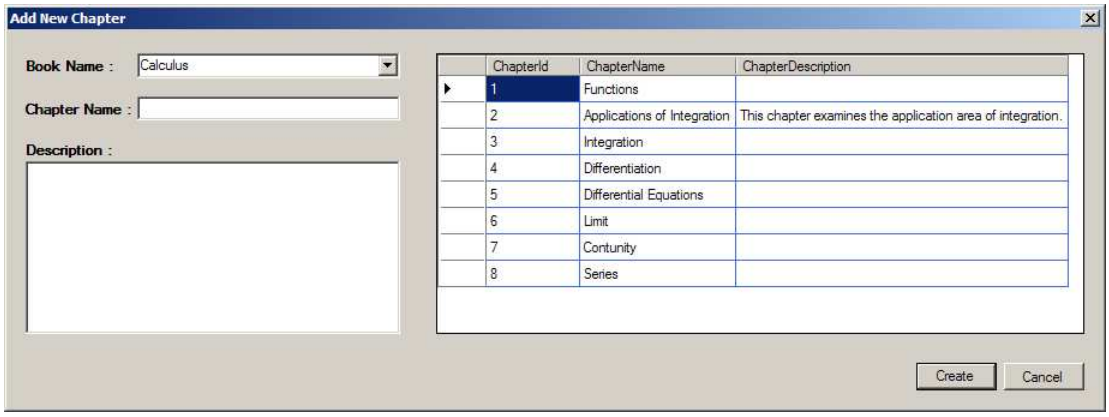


Figure 3.10 Add New Chapter Dialog

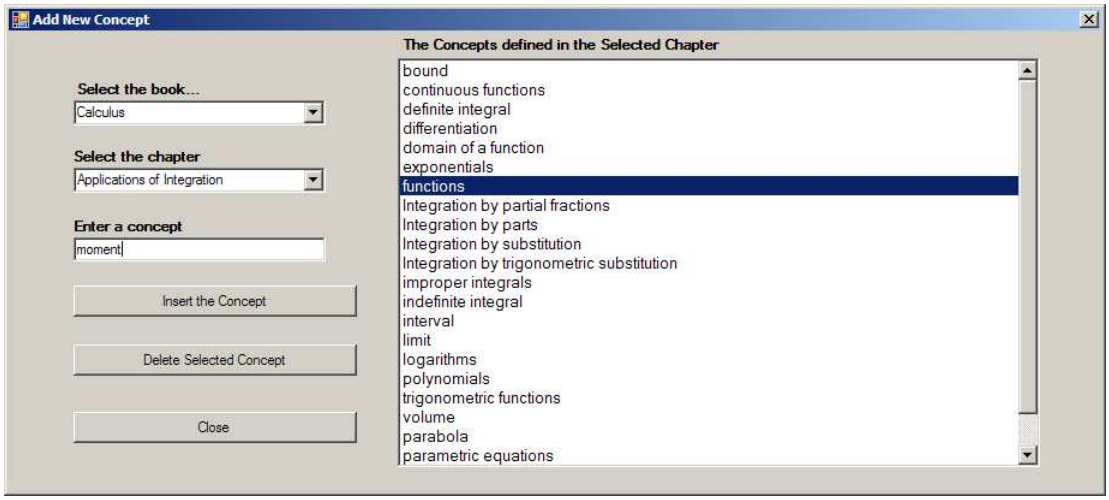


Figure 3.11 Add New Concept Dialog

A bit more complex step is to create the concept map after defining all the concepts. For this prupose user must select the "*Create Concept Map*" command from the "*Tools*

*Menu*". With this command, the teacher can connects the concepts to together and generate the concept map by specifying the relationships between concepts. This is the most important step of this scenario. As seen in Figure 3.12, the user firstly selects a chapter within a book. After that he/she must select a concept, which wants to relate it to others from the "*Select concept* Combobox. When it has been done, all other concepts are listed in the "*Related Concepts* Listbox. If the user relate one of them with the selected concept, it is sufficient to check it. For instance, when the user select the "*Calculus*" Book and the "*Applications of Integration*" chapter, all the concepts are listed in the combobox. Assume that, the user want to relate the "*trigonometric functions*" concept with the other concept in the chapter, he/she can click the checkedbox nearby the concepts "*bound, functions, continuous functions, domain of a function, interval*. Actually, a student must know all these concepts to solve the questions related to concept "*trigonometric functions*". The teacher must relate the concepts with this manner. After the teacher determine the related concepts, he/she clicks the "*Generate Concept Map*" button. Then the system associates the concepts.

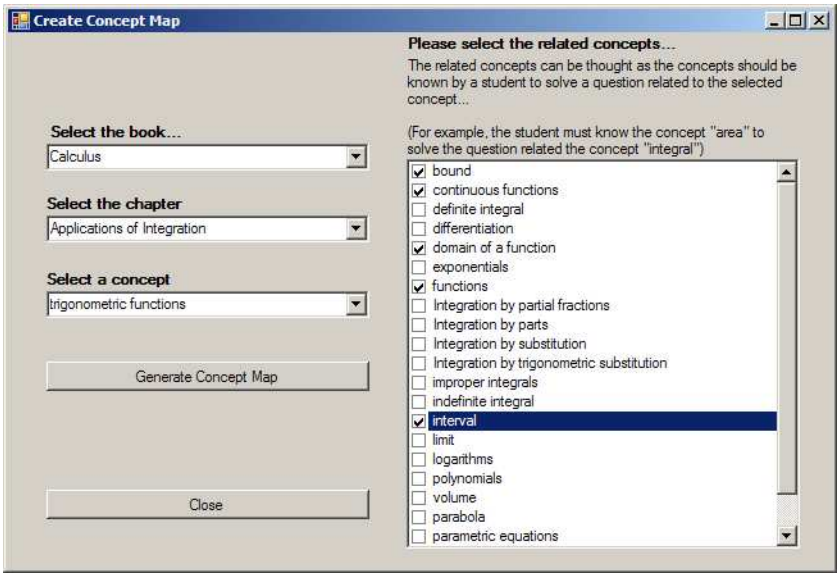


Figure 3.12 Generating concept map

As such in the case of adding a new chapter, the teacher can add a section within a chapter. The process is completely the same. To create a section, teacher must select the "*Add Section*" command from the "*Tools Menu*".

At the next step, user adds a lesson into the section. To do so, he/she must select the "*Add Lesson*" command from the "*Tools Menu*". However, the lecture notes must be written as a small lecture units, and the lecture must be written in LaTeX notation. The teacher can get a help from "*Mathematica Kernel* as stated before. If user wants to preview his/her lecture notes, it is enough to click to "*PDF Viewer*" button on the "*Toolbox*". Finally, to add the written notes into the MathITS, it must be selected the "*Add Lesson* command from "*Tools Menu*". After that a "*Select Dialog Box*" is viewed as in Figure 3.13 and the user selects the section, so that the lesson can added.

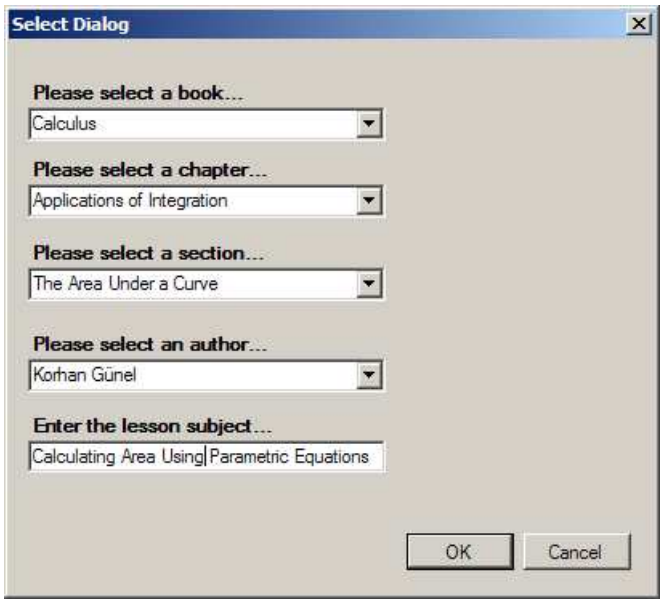


Figure 3.13 Select Dialog Form

The final step is the adding questions related to a lesson and associating the question with concepts. To do this, user must select the "*New Question*" command from the "*File Menu* and fill the blanks with LaTeX code. After that, he/she must select the '*Add Question*" command from the "*Tools Menu*". All the process is the same as

adding lesson into the system. After related lesson is selected from the "*Select Dialog Box*", the message "Your question has been added to system. Now, please specify the relationship between your question and the concepts!" is given as a feedback to user. When the user "*Ok*" button, the "Test Item Relationship Table(TIRT)" form is opened as seen in Figure 3.14.

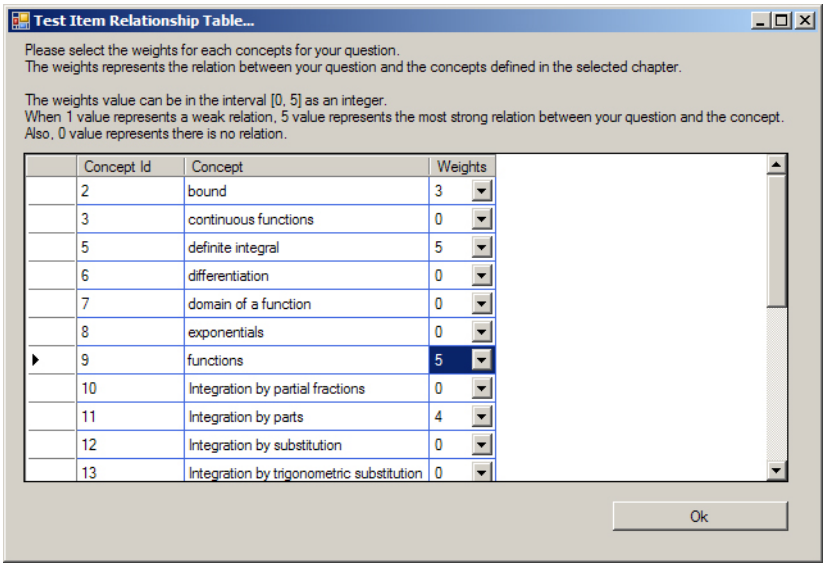


Figure 3.14 Test Item Relationship Table

With the form as depicted in Figure 3.14, user specify the relationships between the added question and the concepts defined in the chapter. The relation between the question and a concept is determined by a weight value. The weights value can be in the interval $[0, 5]$ as an integer. Whereas 1 value represents a weak relation, 5 value represents the most strong relation between your question and the concept. Also, 0 value represents there is no relation. After the user assign weights values for each concept and then click "*Ok*" Button, the difficulty rate of the question is calculated by the system as seen in Figure 3.15. Thus, the all jobs have been done by the teacher.

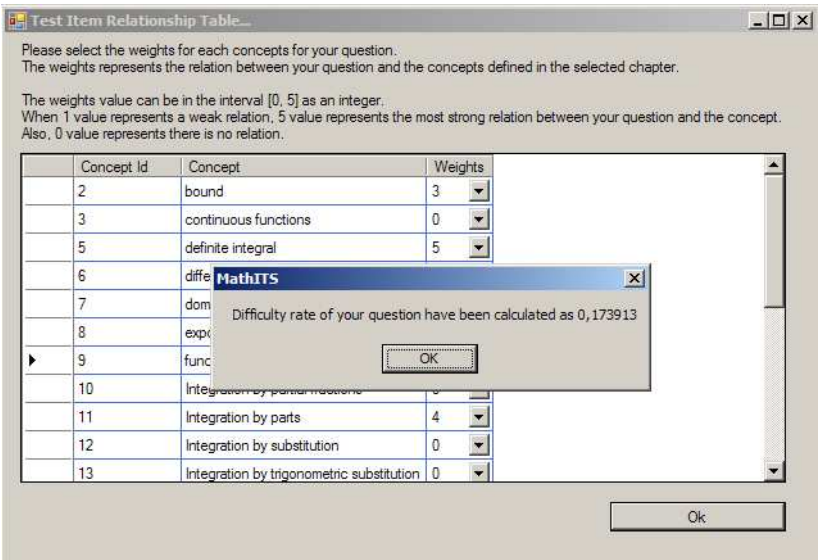


Figure 3.15 Calculating question difficulty rate

## 3.3 User Guide for Students

Although the student users is the central of the system, the scenario is so simple for students. The student selects a lesson to study and takes a test. When a user logs into the system as a student, the Administration Module calls the "*Student Module Form*" as shown in Figure 3.16. All the chapters of the selected book placed in knowledge-base are listed, when the form is loaded. If user selects a chapter from the listbox, then all the sections related to the selected chapter, are listed and the description of chapter is summarized. Similarly, if the user selects a section from the listbox, the lectures in the knowledgebase of the selected section are added to the listbox and the section description is given.

Finally, when student selects a lecture form the list, then the author of the lecture is written. This is the one of the advantage of using MathITS. Because student can simply select to study one of the same lecture unit written by different authors. If he/she select one of the lesson from the listbox, then the *Start Lesson* button is activated. After *Start Lesson* button is clicked, then the system gets the related lecture

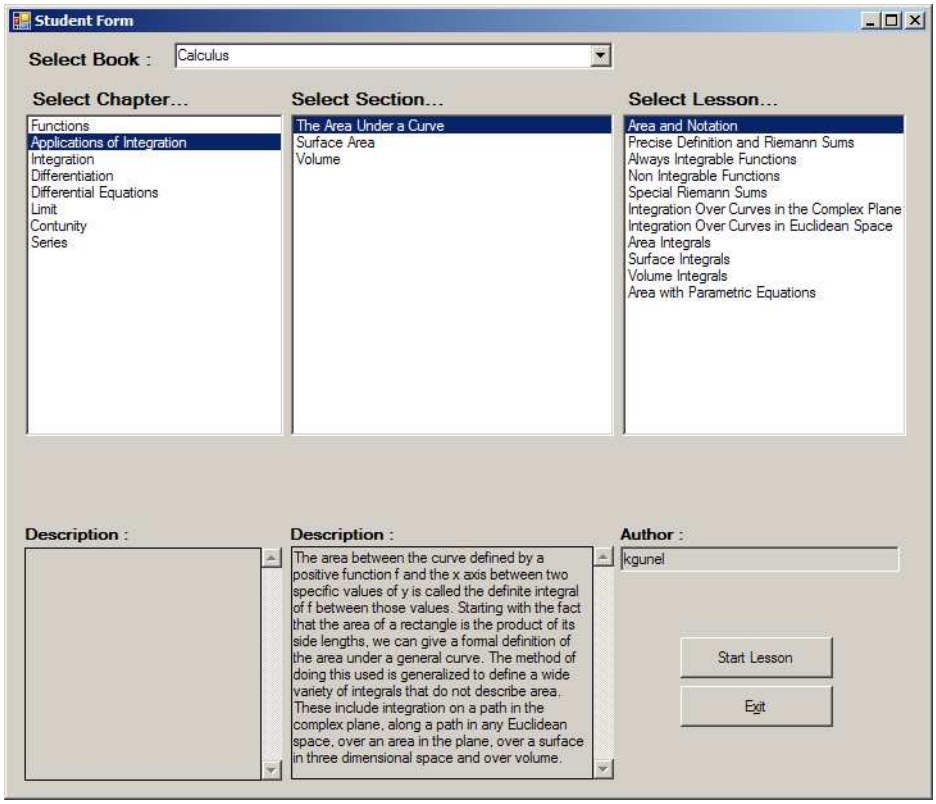


Figure 3.16 The user interface for a student

unit from the knowledgebase in LaTeX notation and compiles the notes with LaTeX. Consequently, a pdf file is generated and then loaded into a new form called by *Sample Lesson* as shown in Figure 3.17.

On the *Sample Lesson* form, there is a button, which is used to generate a multiple choices question test related to the selected lesson. When it is clicked, totally 10 questions are randomly selected from the knowledgebase, even there are more than 10 questions related to the selected lesson. Thus, every time a different test can be generated. The selected questions are asked to student orderly, shown in Figure 3.18.

At the same time, the student can take a hint or get the solution for each questions, after answering. At the time to clicked "*Hint*" button, the PDF file dynamically changed and the related hint is added to the end of question as depicted in Figure 3.19.

If student selects the one of the choices and then clicks the button "*Check answer*", then the Expert Module checks whether the answer is correct or not. If the student fails to answer, then the Expert Module suggests the student to see the solution of the question. If the student wants to check out the solution, the PDF File is automatically changed and the solution is added at the end of the question as seen in Figure 3.20.

When student completes the test, his/her performance is calculated by the Math-ITS as introduced in Chapter 2. The related Concept Effect Table (CET), Test Item Relationship Table (TIRT) and Question Difficulty Rates Table in MathITS are given Appendix Four. The feedback given to the student is the learning percent of related concepts of the test. The output of the system can be seen in Figure 3.21

Therefore, MathITS determines the students weaknesses on concepts using conceptual map modelling. Also, the system advices to the student to study the misunderstood concepts determined by the Concept Effect Table.

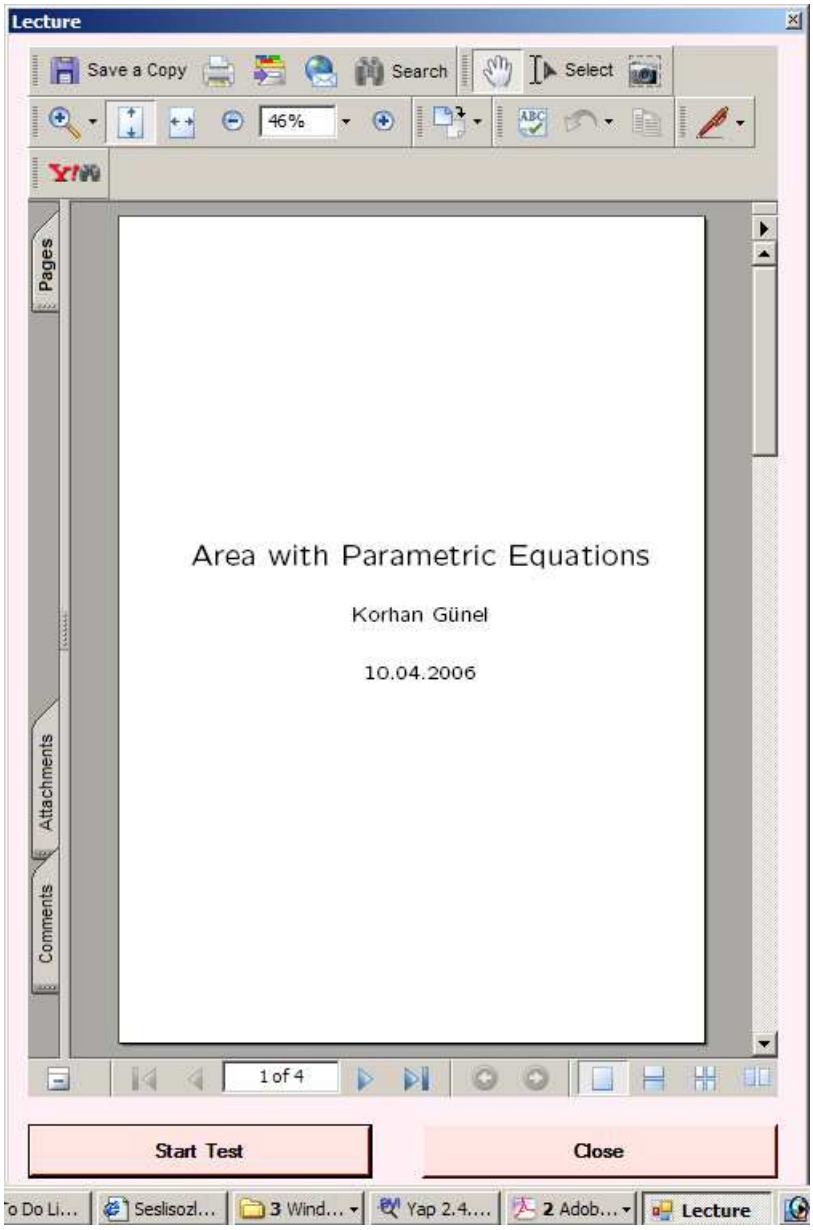


Figure 3.17 Sample Lesson Form

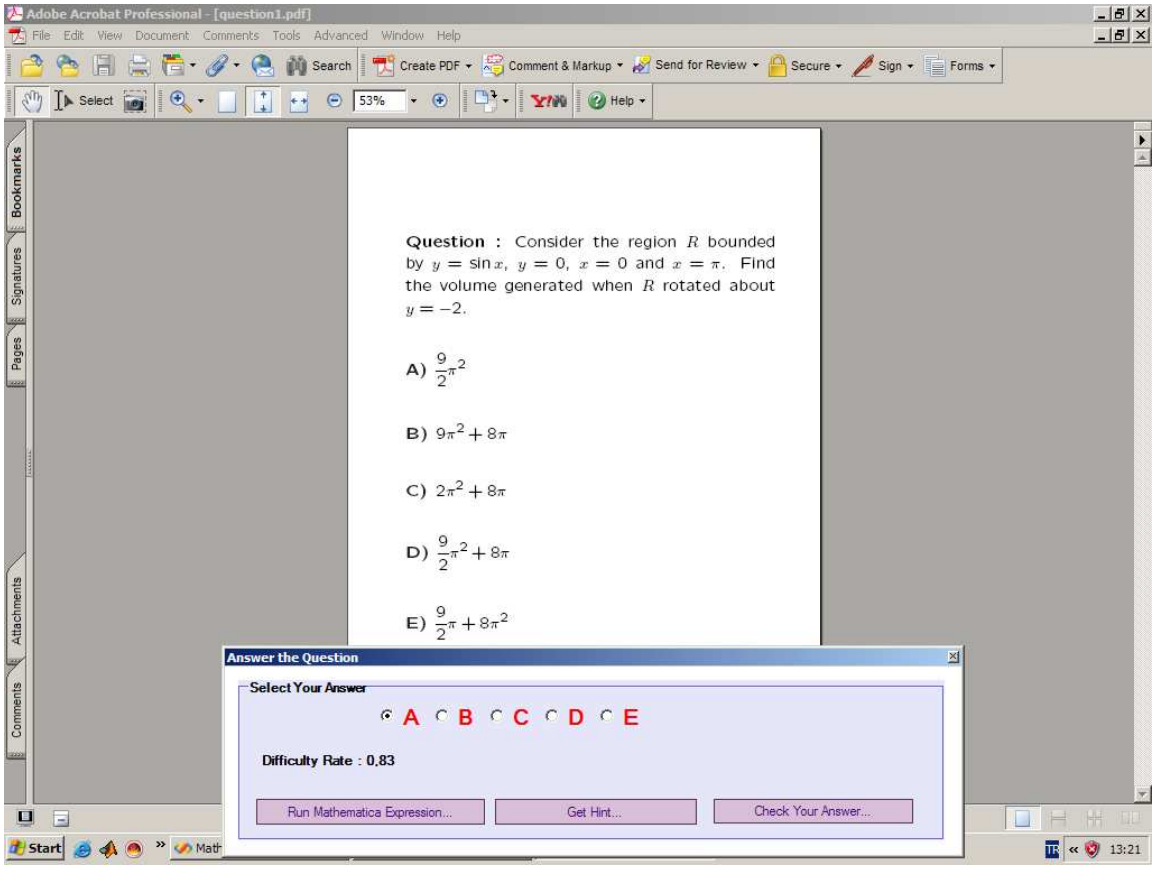


Figure 3.18 Randomly selected test question

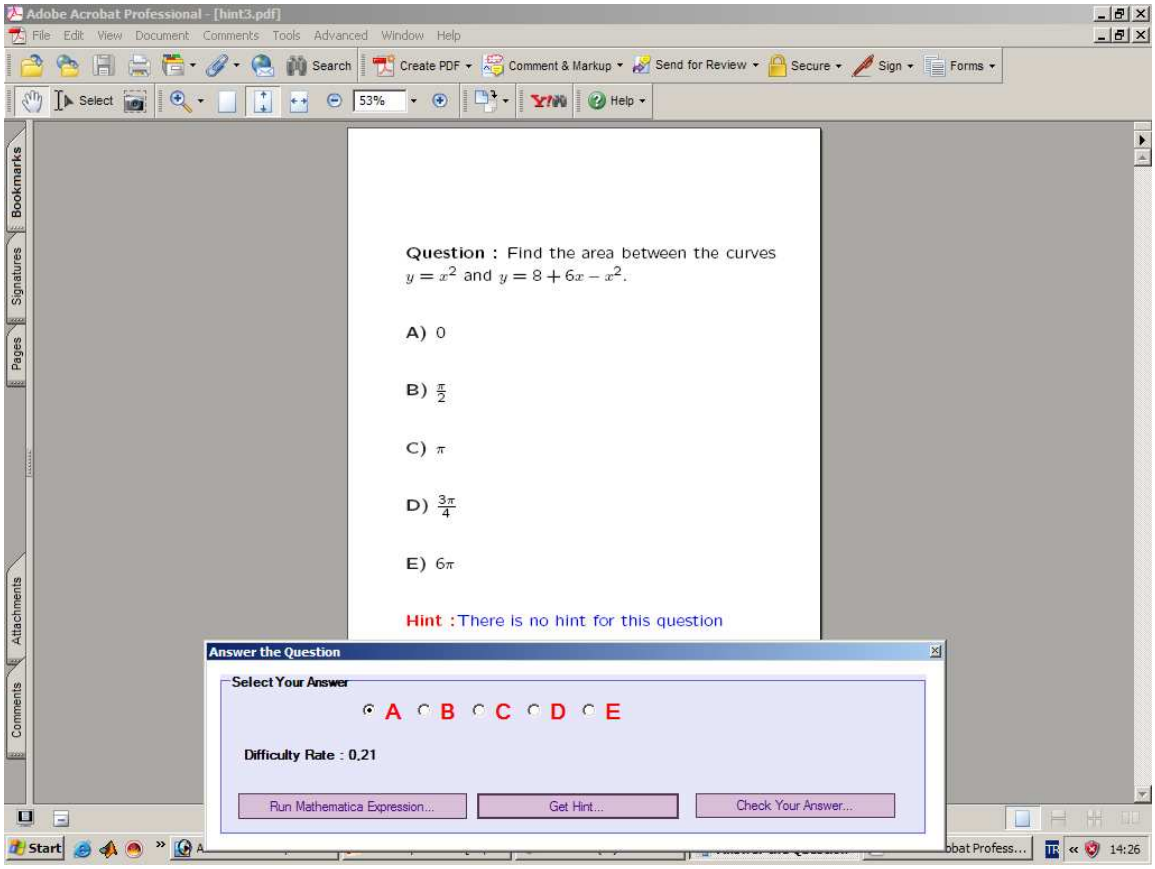


Figure 3.19 Hint for a test question

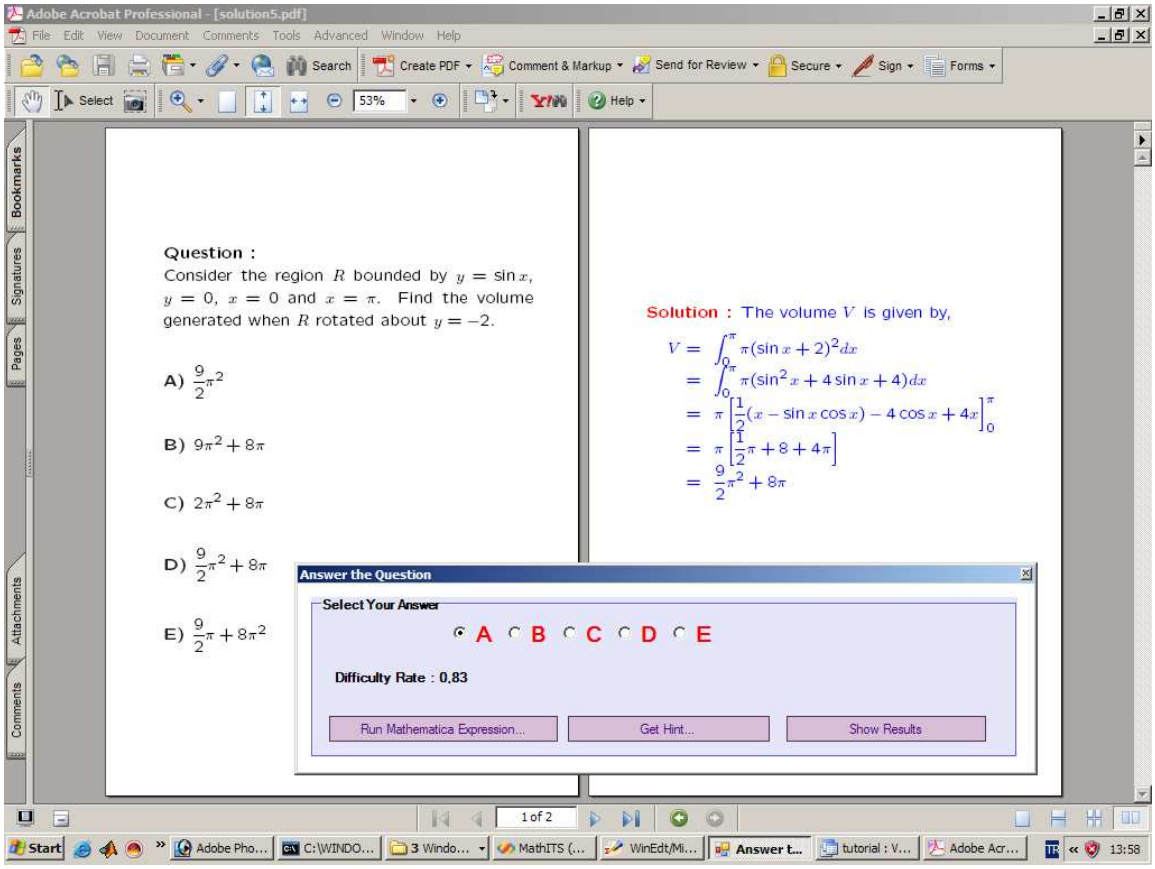


Figure 3.20 Solution of a test question

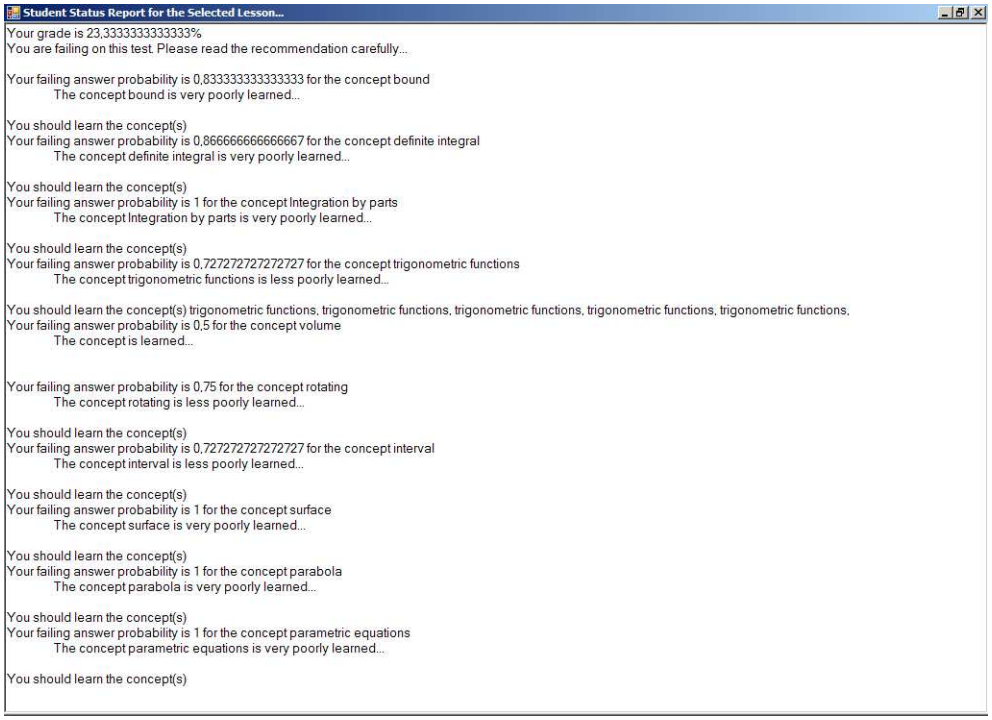


Figure 3.21 The feedback providing to student after a test

# CHAPTER FOUR
## CONCLUSIONS

### 4.1 Concluding Remarks

ITSs are designed to individualise the educational experience of students according to their level of knowledge and skill.

Comparing with classroom teaching, Intelligent Tutoring Systems supports the progress to students performance additionally (Bloom 1984; Anderson, Corbett, Koedinger and Pelletier 1995).

Developing an intelligent tutoring system clearly needs to consider various factors, such as domain, knowledge representation, measuring student performance, preparing lecture notes and feasible tests. They have high development costs. With this manner, this thesis focuses only a small parts of the system.

In this thesis, we proposed an intelligent tutoring system for mathematics education, which uses conceptual map modelling as a student modelling paradigm. The knowledge representation used in the system is based on LaTeX notation to write, easily, the mathematical expressions such as sum, integral etc. From viewpoint of an expert user, Mathematica Kernel is used as an expert system. With Mathematica Kernel, an expert can easily write questions, solve them and he/she simply adds to the system in LaTeX representation.

To illustration of the system, the domain was selected as small as possible in the thesis. The only one lecture unit has been added into the knowledgebase is ”Calculating Area with Parametric Equations”, which its knowledge representation is as given Appendix Two. The knowledgebase stores 24 different concepts as seen in Appendix

Four and a sample test, involves totally 11 questions, is shown in Appendix Three. Also the related CET, QDRT and TIRT tables in knowledgebase of MathITS are given in Appendix Four.

Based on the proposed model, MathITS system is implemented which can identify poorly-learned and well-learned concepts for individual students.

## 4.2 Future Work

The researches of ITSs have only produced hopes and some prototypes, however these prototypes have failed to prove the usefulness in academic environment (Kinshuk, 2002). Thus, some questions arise: “How can ITSs become powerful tools in the hands of teachers?”, “Could it be possible to replace with teachers and ITSs?”, “Does intelligent tutoring have future?” or “What are the future works for progressing ITS?”.

With the current methodology of developing ITS, it is seen that the prototypes of ITS have been developed independently and tutoring expertise were hard-coded into individual applications, so far.

It is necessity to developed reusable components, such as the student model, tutoring model, and user interface. A standard language for representing knowledge and a set of tools to manipulate the knowledge must be generated. Eventually, these systems must be developed as general-purpose software tools such as word processors and spreadsheets. In this way, they can be independent from the domain. With the viewpoint of mathematics education at graduate level, ITS should be capable of proving theorems.

Finally the system should support Natural Language Processing(NLP) and Speech Recognition issues. Thus, the computer acts as a real teacher and it provides more effective one-to-one learning for each of students.

**REFERENCES**

Amižić A., Stankov S., & Rosić M. (2002), Model tracing – a diagnostic technique in intelligent tutoring systems. In *Proceedings of the (Central European Exchange Programme for University Studies) CEEPUS Summer School Jointly with the 5th Symposium on Intelligent Systems*, Split, Croatia, iSBN 953-96516-8-9.

Devedzic V., & Harrer A. (2005), Software patterns in its architectures. In *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 15, pp. 63–94.

Hafner K. (2000), Intelligent tutoring systems (a subtopic of education).

Harp S.A., Samad T., & Villano M. (1995), Modelling student knowledge with self-organizing feature maps. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(5), 727–737.

Hwang G.J. (2003), A conceptual map model for developing intelligent tutoring systems. *Computers and Education*, 40, 217–235, this study is supported by the National Science Council of Republic of China.

Jeremic Z., & Devedzic V. (2004), Design pattern its: Student model implementation. In *ICALT*.

Kai-min Chang Joseph Beck J.M.A.C. (2006), A bayes net toolkit for student modeling in intelligent tutoring systems. In $8^{th}$ *International Conference on Intelligent Tutoring Systems*.

Karlgren K. (2005), Intelligent tutoring systems (its). Retrieved March 9, 2006, from http://www.dsv.su.se/ klas/Learn/ITS/its.html.

Kinshuk A.P. (2002), Does intelligent tutoring have future! In *ICCE'02 IEEE Proceedings of the International Conference on Computers in Education*.

Martin B., & Mitrovic A. (2001), Easing the its bottleneck with constraint-based modelling. *New Zealand Journal of Computing*, 8(3), 38–47.

Martin B.I. (2001), *Intelligent Tutoring Systems: The Practical Implementation of Constraint-based Modelling*. Ph.D. thesis, University of Canterbury.

Molnar A. (1997), Computers in education: A brief history. *THE Journal*, 24, retrieved January 14, 2006, from http://www.thejournal.com/articles/13739.

Rehani B., & Sasikumar M. (2003), Chaatra: A student monitoring and learner modelling system. This work was carried out under the Vidyakash project of the Department of Information Technology, Ministry of Communications and Information Technology, Goverment of India.

Self J. (1990), Theoretical foundations for intelligent tutoring systems. Tech. Rep. AAI/AI-ED Technical Report No.45 (in Journal of Artificial Intelligence in Education, 1(4), 3-14), Computing Department, Lancaster University.

Shang Y., Shi H., & Chen S.S. (2001), An intelligent distributed environment for active learning. *ACM Journal of Educational Resources in Computing*, 1(2), 4.

The TeX Users Group TUG (2006), Just what is tex?. available on http://www.tug.org/whatis.html, received at 06.05.2006.

Vanlehn K. (2001), Bayesian student modeling, user interfaces and feedback: Sensitivity analysis. *International Journal of Artificial Intelligence in Education*, 12(2), 154–184.

Wolfram S. (2002), *The Mathematica Book*. Wolfram Research.

# APPENDIX ONE
## SYSTEM REQUIREMENTS

Table A.1 System Requirements for MathITS

| Component | Required | Recommended |
|---|---|---|
| **Operating System** | Microsoft Windows® 98 Second Edition, Windows 2000, Windows Millennium Edition, Windows XP Home Edition, or Windows XP Professional | Windows XP Home Edition or Windows XP Professional |
| **Processor** | A 233 megahertz (MHz) processor, such as an Intel Pentium II or Advanced Micro Devices (AMD) processor | A 500 MHz processor or faster |
| **RAM** | 64 MB | 256 MB |
| **Free harddisk space** | 100 MB | Approximately 300 MB to use System Restore |
| **Optical Drive** | CD or DVD drive | CD-R or CD-RW drive |
| **Video adapter and monitor** | Super VGA (800 x 600) or higher resolution | Same as minimum configuration |
| **Software** | Adobe Acrobat Reader v6.0, Version 1.1 of the .NET Framework, LaTeX distribution package as MikTeX | Same as minimum configuration |

# APPENDIX TWO

## THE STRUCTURE OF KNOWLEDGEBASE

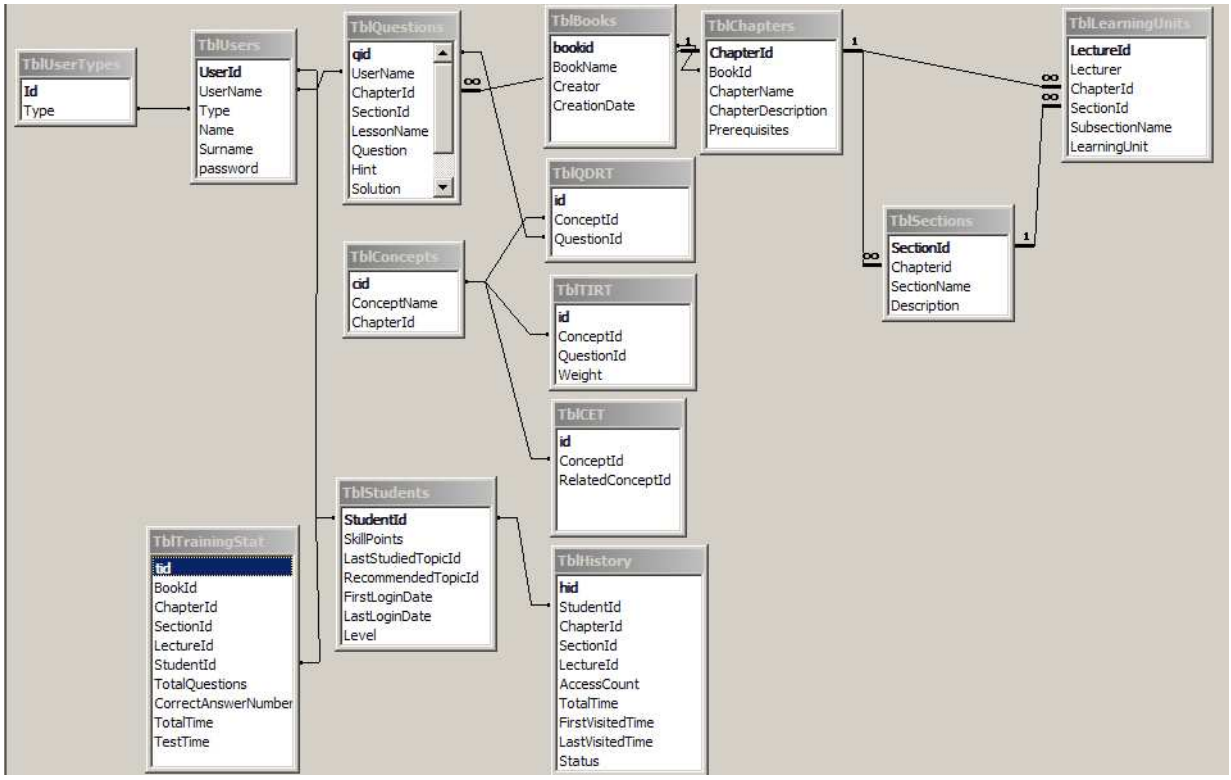


Figure B.1 The structure of knowledgebase and database of MathITS.

# APPENDIX THREE

## SAMPLE LECTURE UNIT, SAMPLE QUESTIONS IN MATHITS

### C.1 Sample Lecture Unit

```
\begin{lesson}

\title{Area with Parametric Equations}
\author{Korhan G\u nel}
\date{10.04.2006}
\maketitle

\begin{slide}

In this section we will find a formula for determining the area
under a parametric curve given by the parametric equations,
\textcolor[rgb]{0.00,0.50,1.00}{$$
\begin{array}{l}
x=f(t)\\
y=g(t)\\
\end{array}
$$}

We will also need to further add in the assumption that the curve
is traced out exactly once as t increases from
\textcolor[rgb]{0.00,0.50,1.00}{$\alpha$} to
\textcolor[rgb]{0.00,0.50,1.00}{$\beta$}.
```

We will do this in much the same way that we found the first derivative in the previous section. We will first recall how to find the area under \textcolor[rgb]{0.00,0.50,1.00}{$y=F(x)$} on \textcolor[rgb]{0.00,0.50,1.00}{$a\leq x \leq b$}.
\textcolor[rgb]{1.00,0.50,0.50}{$$
\displaystyle{A=\int_{a}^{b}F(x)dx}
$$}

We will now think of the parametric equation \textcolor[rgb]{0.00,0.50,1.00}{$x=f(t)$} as a substitution in the integral. We will also assume that \textcolor[rgb]{0.00,0.50,1.00}{$a=f(\alpha)$} \quad and \quad \textcolor[rgb]{0.00,0.50,1.00}{$b=f(\beta)$} for the purposes of this formula. There is actually no reason to assume that this will always be the case and so we will give a corresponding formula later if it is the opposite case \quad (\textcolor[rgb]{0.00,0.50,1.00}{$b=f(\alpha)$} \quad and \quad \textcolor[rgb]{0.00,0.50,1.00}{$a=f(\beta)$}).

So, if this is going to be a substitution we will need,
\textcolor[rgb]{1.00,0.50,0.50}{$$ dx=f’(t)dt
$$}

Plugging this into the area formula above and making sure to change the limits to their corresponding $t$ values gives us,
\textcolor[rgb]{1.00,0.50,0.50}{$$
\displaystyle{A=\int_{\alpha}^{\beta}F(f(t))f’(t)dt}
$$}

```
Since we do not know what \textcolor[rgb]{0.00,0.50,1.00}{$F(x)$}
is we will use the fact that \textcolor[rgb]{1.00,0.50,0.50}{$$
y=F(x)=F(f(t))=g(t)
$$}
and we arrive at the formula that we want.
\textcolor[rgb]{1.00,0.50,0.50}{$$
\displaystyle{A=\int_{\alpha}^{\beta}g(t)f'(t)dt}
$$}
\end{slide}
\end{lesson}
```

## C.2 Sample Questions

**$Q_1$** : Determine the area under the parametric curve given by the following parametric equations.

$$x = 6(\theta - \sin\theta), y = 6(1 - \cos\theta), \qquad 0 \leqslant \theta \leqslant 2\pi.$$

**$Q_2$** : Find the area between the curves $y = x^2$ and $y = 2x + 3$.

**$Q_3$** : Find the area of the region bounded vertically by $y = x^2$ and $y = x + 2$ and bounded horizontally by $x = -1$ and $x = 3$.

**$Q_4$** : Find the area between the curves $y = x^2$ and $y = 8 + 6x - x^2$.

**$Q_5$** : Find the area of the surface obtained by rotating the curve $y = \frac{1}{4}(e^{2x} + e^{-2x})$ with $0 \leqslant x \leqslant 1$.

**$Q_6$** : Just set up the integral for the surface obtained by rotating the curve $y = \frac{1}{4}(e^{2x} + e^{-2x})$ with $0 \leqslant x \leqslant 1$ around the y-axis.

**$Q_7$** : Determine the surface area of the solid obtained by rotating $y = \sqrt{9 - x^2}$, $-2 \leqslant x \leqslant 2$ about the x-axis.

$\mathbf{Q_8}$ : Consider the region $R$ bounded by $y = sinx$, $y = 0$, $x = 0$ and $x = \pi$. Find the volume generated when $R$ rotated about x-axis.

$\mathbf{Q_9}$ : Consider the region $R$ bounded by $y = sinx$, $y = 0$, $x = 0$ and $x = \pi$. Find the volume generated when $R$ rotated about y-axis.

$\mathbf{Q_{10}}$: Consider the region $R$ bounded by $y = sinx$, $y = 0$, $x = 0$ and $x = \pi$. Find the volume generated when $R$ rotated about $y = -2$.

$\mathbf{Q_{11}}$ : Consider the region $R$ bounded by $y = sinx$, $y = 0$, $x = 0$ and $x = \pi$. Find the volume generated when $R$ rotated about $x = 2\pi$.

# APPENDIX FOUR

## SAMPLE CONCEPTS AND QDRT, CET, TIRT IN MATHITS

Table D.1 Some Concepts Defined in MathITS

| Id | Concept | Id | Concept |
|---|---|---|---|
| $C_1$ | area | $C_{13}$ | interval |
| $C_2$ | bound | $C_{14}$ | limit |
| $C_3$ | continuous functions | $C_{15}$ | logarithms |
| $C_4$ | definite integral | $C_{16}$ | polynomials |
| $C_5$ | differentiation | $C_{17}$ | indefinite integral |
| $C_6$ | domain of a function | $C_{18}$ | improper integrals |
| $C_7$ | exponentials | $C_{19}$ | trigonometric functions |
| $C_8$ | functions | $C_{20}$ | volume |
| $C_9$ | Integration by partial fractions | $C_{21}$ | parabola |
| $C_{10}$ | Integration by parts | $C_{22}$ | parametric equations |
| $C_{11}$ | Integration by substitution | $C_{23}$ | surface |
| $C_{12}$ | Integration by trigonometric substitution | $C_{24}$ | rotating |

Table D.2 The Question Difficulty Rate Table for the Sample Test in MathITS

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ | $C_{16}$ | $C_{17}$ | $C_{18}$ | $C_{19}$ | $C_{20}$ | $C_{21}$ | $C_{22}$ | $C_{23}$ | $C_{24}$ | $Nc$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_1$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 4 |
| $Q_2$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 |
| $Q_3$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 |
| $Q_4$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| $Q_5$ | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| $Q_6$ | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| $Q_7$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 6 |
| $Q_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 3 |
| $Q_9$ | 1 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 7 |
| $Q_{10}$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 7 |
| $Q_{11}$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 7 |

Table D.3 The Concept Effect Table in MathITS

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ | $C_{16}$ | $C_{17}$ | $C_{18}$ | $C_{19}$ | $C_{20}$ | $C_{21}$ | $C_{22}$ | $C_{23}$ | $C_{24}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $C_2$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_3$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_4$ | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $C_5$ | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_7$ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_8$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_9$ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{10}$ | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{11}$ | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{12}$ | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{13}$ | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{14}$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{15}$ | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{16}$ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{17}$ | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{18}$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{19}$ | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{20}$ | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $C_{21}$ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{22}$ | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $C_{23}$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| $C_{24}$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| $NP_j$ | 2 | 15 | 10 | 10 | 8 | 20 | 3 | 23 | 4 | 4 | 5 | 4 | 20 | 4 | 4 | 3 | 4 | 0 | 7 | 2 | 0 | 5 | 1 | 1 |

Table D.4 Test Item Relationship Table(TIRT) costructed in MathITS

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ | $C_{16}$ | $C_{17}$ | $C_{18}$ | $C_{19}$ | $C_{20}$ | $C_{21}$ | $C_{22}$ | $C_{23}$ | $C_{24}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_1$ | 5 | 2 | 4 | 3 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 5 | 0 | 0 |
| $Q_2$ | 5 | 5 | 0 | 3 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| $Q_3$ | 5 | 5 | 0 | 3 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| $Q_4$ | 5 | 5 | 0 | 3 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| $Q_5$ | 5 | 2 | 0 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 5 |
| $Q_6$ | 5 | 2 | 0 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 5 |
| $Q_7$ | 5 | 2 | 0 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 5 |
| $Q_8$ | 0 | 2 | 0 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 0 | 0 | 4 | 5 |
| $Q_9$ | 0 | 2 | 0 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 0 | 0 | 4 | 5 |
| $Q_{10}$ | 0 | 2 | 0 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 0 | 0 | 4 | 5 |
| $Q_{11}$ | 0 | 2 | 0 | 4 | 0 | 0 | 5 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 0 | 0 | 4 | 5 |
| $Sum(C_j)$ | 35 | 31 | 4 | 40 | 4 | 0 | 35 | 7 | 0 | 0 | 12 | 3 | 22 | 0 | 0 | 0 | 0 | 0 | 24 | 20 | 10 | 5 | 28 | 35 |