DETERMINING DIFFICULTY OF QUESTIONS IN INTELLIGENT TUTORING SYSTEMS

Korhan GÜNEL
Mathematics Department
Ege University
kgunel@gmail.com

Rıfat AŞLIYAN
Computer Engineering Department
Dokuz Eylul University
rasliyan@yahoo.com

ABSTRACT
The object of this study is to model the level of a question difficulty by a differential equation at a pre-specified domain knowledge, to be used in an educational support system. For this purpose, we have developed an intelligent tutoring system for mathematics education. Intelligent Tutoring Systems are computer systems designed for improvement of learning and teaching processes in the domain knowledge. The developed system, which is called as MathITS, is based on conceptual map modeling.The Mathematica Kernel is used as an expert system and knowledge representation is based on *LaTeX* notation in MathITS.

1. INTRODUCTION

Nowadays, it is too hard to imagine the education without computers. When properly applied, computer assisted educational technologies can provide effective means for learning. In this manner, we have developed an intelligent tutoring system to teach mathematics at undergraduate and graduate levels.

An intelligent tutoring system (ITS) is used to enable the students work independently, to improve their understanding of concepts within the related domain, and to observe progress of problem solving ability. Of course, an ITS can assist not only to the students but also to the teachers for developing and managing the courses.

Mathematics is a nightmare for lots of students. Inevitably, many students doubt their intelligence, creativity, talent, and motivation when studying mathematics. In class based education, the real teachers which have some mental capabilities such as the reasoning ability, planning, problem solving, abstract thinking, comprehending ideas, and learning, can motivate and encourage the students with intitutively selected methods. They, also take the creativity, personality, or character of students into account. In this sense, the tutoring systems must have the capability of real teachers as much as possible. It should permanetly encourage the students to study and keep them in high motivation. To unbreak their enthusiasm, the system must ask different questions according to the level and capability of each student. Hence, determining the difficulty of a question to test the understanding of the related concepts is crucial.

Intelligent Tutoring Systems can statistically model understanding the concepts in the learning domain. Several methods have been proposed to determine the hardness of questions (Khan et al., 2003, Kunichika et al., 2002, Li & Sambasivam, 2003). 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. Originally, concept maps should be thought as directed graphs. According to this method, each question consists of some concepts, and Hwang specifies the hardness of questions related to the number of concepts to be learned. In these studies, the hardness of questions depends on only the number of related concepts, and these approaches take into account neither the learning perfomance of students nor the possibility of inefficient training.

In this paper, a population dynamics based model has been proposed which models hardness of questions using differential equations. The developed model has been applied to mathematics education. An ITS has been developed which uses the conceptual map modeling technique and in which the hardness of the question to be asked to students is determined by a dynamical equation. This enables the tutors to track the learning performance of students individually.

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 (Kinshuk, 2001). They have high development costs. With this manner, this study focuses on only a small part of the system. In the sequel, firstly the student modeling which uses the conceptual map modeling approach, has been explained. Then, the population dynamic based determination of questions’ hardness procedure has been accomplished. Finally, MathITS system has been introduced as the implementation of these concepts.

2. STUDENT MODELLING

The most critical component of ITS is the student module whose necessity has been addressed by Self (1999). Simply, this module is about the theory of student behaviours, and it generates all information about individual learner. The student model evaluates each learner's performance to determine his or her knowledge, perceptual abilities and reasoning skills. It provides the information such as what the student knows or does not know, any misconceptions, student's degree of forgetfulness (Jeremic & Devedzic, 2004).

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. Figure 1 simply illustrates the concept map for the "number" concept.

The conceptual map approach offers a 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 suggests 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 different 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.

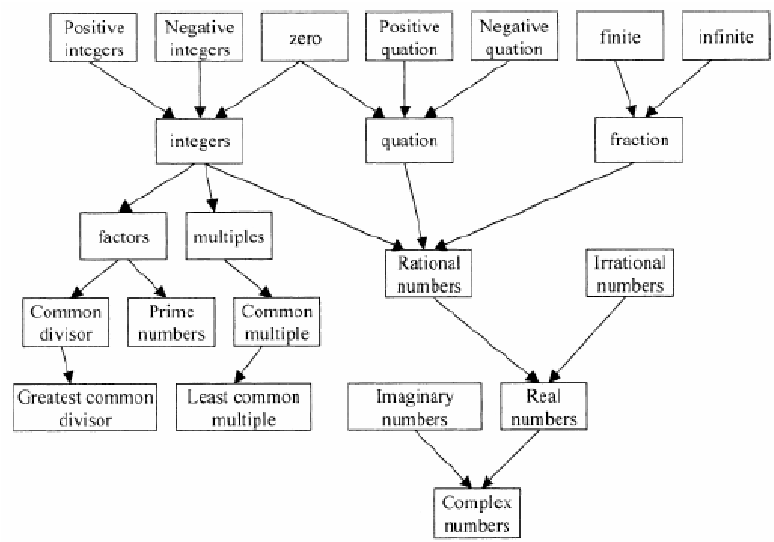

Figure 1. Concept map for numbers (Hwang, 2003).

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 1 demonstrates how the CET is constructed. As seen in Table 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 demonstrates the difficulty rates for each question. Initially, 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 $10 \times Nc_i / n$. According to Table 2, the test contains 10 questions and *n* different concepts. Thus, the QDRT value of the question Q_5 is QDRT(5) = 3/*n* and the mark of this question is $10 \times 3/n$ points at the first step. 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 questions and *n* different concepts. Then total mark for this test is calculated in Equation 1.

$$Total\ Mark = 10 \times \frac{\sum_{i=1}^{m} Nc_i}{n} \qquad (1)$$

Table 1. An Example of the Concept Effect Table

	C_1	C_2	C_3	…	C_j	…	C_n
C_1	0	0	1	…	0	…	0
C_2	0	1	1	…	0	…	1
C_3	0	0	0	…	1	…	1
⋮	⋮	⋮	⋮		⋮		⋮
C_j	0	0	0	…	1	…	1
⋮	⋮	⋮	⋮		⋮		⋮
C_n	0	0	0	…	0	…	0
NP_j	0	1	2	…	2	…	3

Table 2. The Question Difficulty Rate Table (QDRT)

	C_1	C_2	C_3	…	C_j	…	C_n	Nc
Q_1	1	1	0	…	0	…	0	6
Q_2	0	1	1	…	0	…	0	5
Q_3	1	0	1	…	1	…	0	4
Q_4	0	0	0	…	1	…	1	4
Q_5	1	0	0	…	0	…	0	3
Q_6	1	1	0	…	1	…	0	3
Q_7	0	1	1	…	0	…	0	2
Q_8	0	1	0	…	1	…	0	7
Q_9	1	1	0	…	1	…	0	4
Q_{10}	1	1	0	…	0	…	1	4

Table 3. The Test Item Relationship Table (TIRT)

	C_1	C_2	C_3	…	C_j	…	C_n
Q_1	5	1	0	…	0	…	0
Q_2	0	4	2	…	0	…	0
Q_3	2	0	3	…	1	…	0
Q_4	0	0	0	…	3	…	5
Q_5	1	0	0	…	0	…	0
Q_6	4	1	0	…	2	…	0
Q_7	0	5	2	…	0	…	0
Q_8	0	0	5	…	2	…	0
Q_9	3	2	0	…	4	…	4
Q_{10}	1	1	0	…	0	…	1
$\sum C_j$	17	14	12	…	12	…	10

Hwang (2003) also proposed the Test Item Relationship Table(TIRT) to calculate the total strength of a concept as shown in Table 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 3, $\sum C_j$ presents the total strength of concept C_j. Briefly, the TIRT table is used to calculate the probability of failure 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 question, then the Equation 2 can be used to calculate failing rates for each concept. Assume that $\mathbf{A}$ represents the student's answers for each of the questions and the test set has m questions. If i^{th} index of $\mathbf{A}$ is 1 then the student answers the question Q_i correctly, otherwise $\mathbf{A}_i = 0$.

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

In Equation 2, $P(C_j)$ represents the probability of failing to answer for the concept C_j. Let $\mathbf{A}$ be $\mathbf{A}=(1, 1, 1, 0, 1, 0, 1, 1, 0, 0)$ for $m = 10$ and $TIRT$ is as given Table 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 can be 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 1, the system can easily determine the related concepts using Concept Effect Table (CET). It suggests that the student should study the concepts C_2, C_3 and C_i.

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 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.

Table 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 <= 0.50	Learned
>= 0 and <= 0.25	Vey well learned

3. DETERMINING THE DIFFICULTY OF QUESTIONS

Although some studies are available in literature, we suggest a new paradigm to determine the level of hardness for a question (Khan et al., 2003, Kunichika et al., 2002, Li & Sambasivam, 2003). As seen in Question Difficulty Rate Table, any question has a value in interval (0, 1] initially. Each time a question has been asked, it has been solved successfully or not by a student. When it occurs, the question difficulty rate must be changed. Motivated by the population dynamics, we propose the following differential equation to recalculate the new difficulty rate of question Q_i, where $\alpha,\beta \in N$ and y represents the new difficulty rate. Originally, this type of differential equation has been used to modeling population dynamics.

$$\begin{cases} y'(t)=\left(\dfrac{\beta-\alpha}{\alpha+\beta}\right)y(t)\left[1-\dfrac{y(t)}{p}\right], & t>t_0 \\ y(t_0)=QDRT(i) \end{cases} \tag{3}$$

In Equation 3, α and β specifies how many times the question Q_i answered correctly and with failure respectively. So, the question Q_i has been asked to students $\alpha+\beta$ times, totally. In the differential equation, $1-\frac{y(t)}{p}$ has been used to stabilize to the system. In this way, when $t\to\infty$, either the solution converges p or it oscillates around p. In particular, we select $p = 1$ to guarantee the solution has the range between [0, 1]. The initial value of the differential equation verifies the condition $0\le y_0\le 1$ and this condition also guarenties $0\le y(t)\le 1$ for all t.

Let $k=\left(\dfrac{\beta-\alpha}{\alpha+\beta}\right)$ and $p=1$, then the solution of the Equation 3 is,

$$y(t)=\frac{y_0}{y_0+\left(1-y_0\right).e^{-kt}} \tag{4}$$

Figure 2 shows the simulation of the changes on the difficulty rate for the question Q_i , which has been answered 49 times correctly and 51 times wrongly. The initial value of *QDRT*(*i*) has been set as 0.5. Each time the question asked, the difficulty rate has been recalculated, according to the value of $\frac{\beta - \alpha}{\alpha + \beta}$.

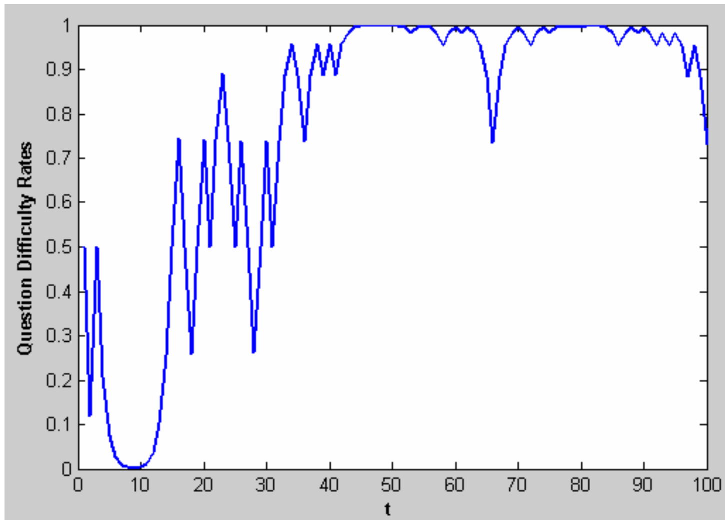

Figure 2. The question difficulty rate with $\alpha = 49$ and $\beta = 51$ and $y_0 = 0.5$.

4. MATHITS : AN INTELLIGENT TUTOR FOR MATHEMATICS EDUCATION

In this study, we suggest an Intelligent Tutoring System for mathematics education, whose architecture can be seen in Figure 3 (Günel, 2006).

MathITS has two types of users. The users login to the system through the administration module. Administration module checks and gets the user information from the database. If the user exists, then the user has been directed to a user interface according to user types.

In addition, MathITS consists of a knowledge representation system, which has strengths unavailable to normal database systems. The knowledge representation systems 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. The knowledge representation in MathITS is based on LaTeX. The original TeX (Tau epsilon Chi) system was built by Donald Knuth. TeX is a computer language designed for using typesetting; in particular, for math and other technical material (TUG, 2006). Although TeX is a relatively low-level language, it is expandable and the common TeX can be combined into macros. The most successful of such macros is called LaTeX, which designed by Leslie Lamport. Hence, MathITS uses LaTeX. In MathITS, all the knowledge such as small lecture units, questions, answers and hints are stored in LaTeX format, and they are converted to PDF documents when it is necessary.

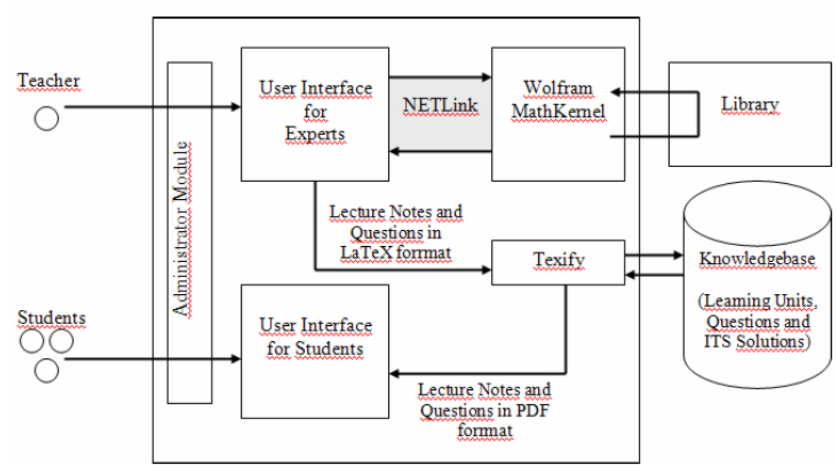

Figure 3. The system architecture

Originally LaTeX is not a word processor! MathITS has 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. All of the knowledge is presented to students with a Portable Document Format (PDF) viewer.

To illustration of this system, the domain has been selected as small as possible in the study. The only lecture unit which has been added into the knowledgebase was "Exact differential equations". The knowledge base stores 24 different concepts and a sample test involving totally 30 questions.

The other module of the system supports a connection to Mathematica Kernel. The Mathematica kernel is basically an interpreter for the Mathematica programming language. This language combines the features of procedural, functional, and rule-based programming together (Wolfram, 2002). The interface provides that an expert can enter Mathematica commands and receives the results in text-based, graphics or LaTeX notation of the command. Figure 4 shows this paradigm.

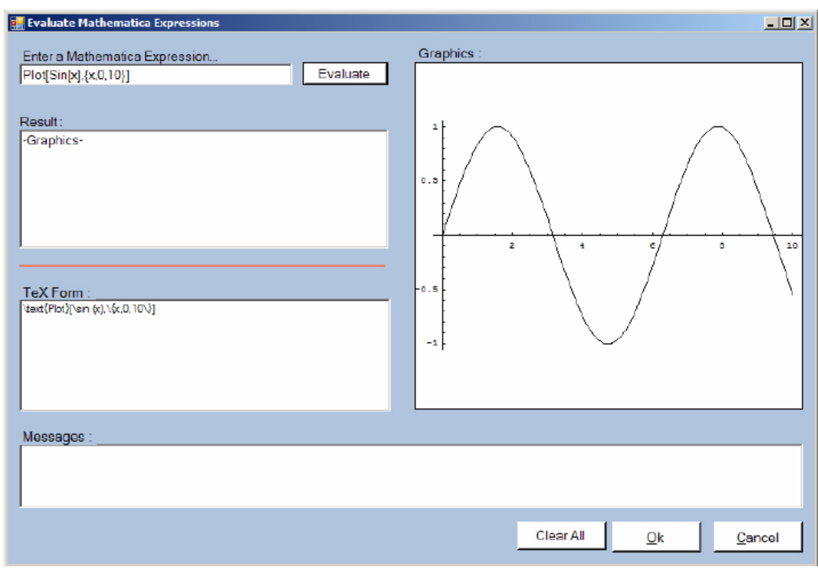

Figure 4. An Interface for Expert Module of system

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 MathITS's user interface and Mathematica Kernel is supported by Microsoft .NET/Link component. Microsoft .NET/Link integrates Mathematica and Microsoft's .NET platform. Microsoft .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. Microsoft .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).

The experts access the user interface shown in Figure 5. Expert users can write small lecture units, examples and questions. If an expert wants to write a 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 of the MathKernel can be seen in two different forms:

- Solution of the expression
- LaTeX notation of expression

The teachers have an essential role in MathITS and have lots of tasks to do. Probably, the most complex scenario within MathITS belongs to the 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 specifies the difficulty level of a question, initially.

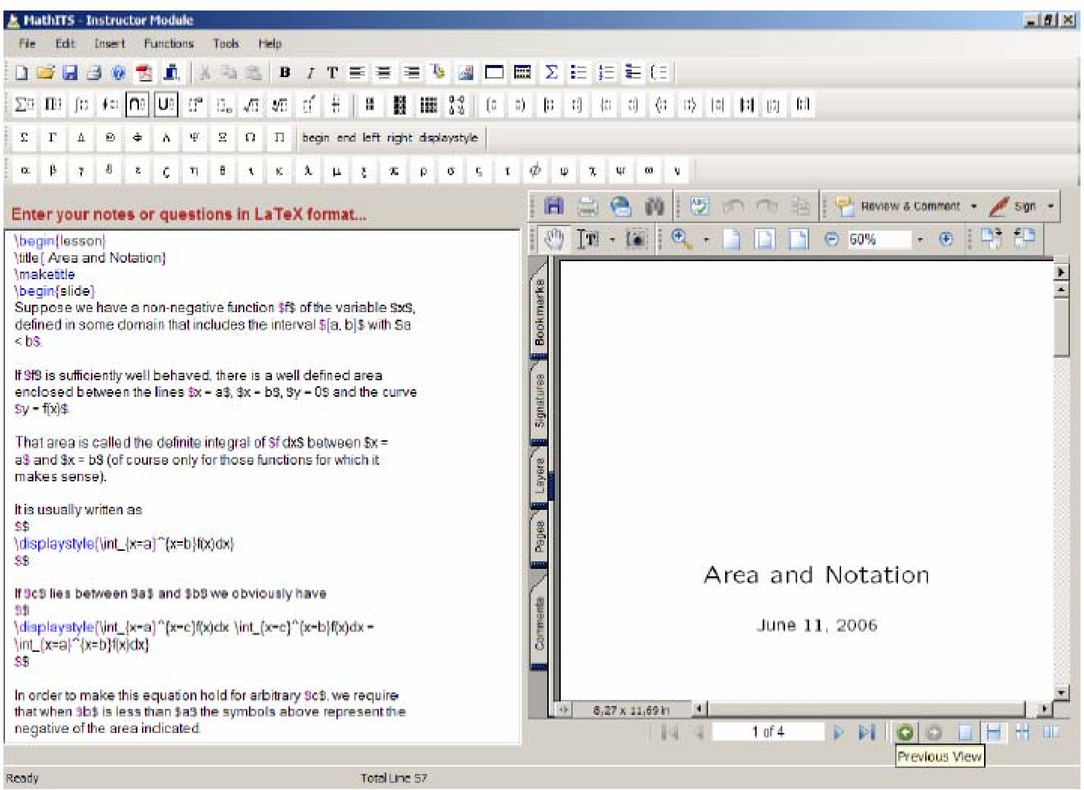

Figure 5. Instructor Module of the system.

Although the students are in the centre of the system, the scenario is so simple for students. The student selects a lesson to study and takes a test. When student completes the test, his/her performance is calculated by the MathITS. Then, the feedback given to the student is the learning rate of the related concepts in the test.

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

5. CONCLUSION

Taking the necessities of the high cost for developing an ITS into account, this study focuses on only the student modeling. We proposed an intelligent tutoring system for mathematics education, which uses conceptual map modelling as a student modeling paradigm. The knowledge representation used in the system is based on LaTeX notation to write the mathematical expressions easily. From the 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.

Based on the proposed model, MathITS can identify poorly-learned and well-learned concepts for individual students. Overall, the main contribution of us is determining the difficulty of questions in ITSs with a new approach. We believe the importance of keeping the students in high motivation in learning and selecting the suitable questions within randomly generated tests or quizzes. The next step of our study will be testing the system with actual students and teachers and observing how it affects their learning performance.

REFERENCES

Günel, K., Intelligent Tutoring Systems for Education, Msc. Thesis, Dokuz Eylul University, Graduate School of Natural and Applied Sciences, İzmir, 2006.

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. *Proceedings. IEEE International Conference on Advanced Learning Technologies,*pp. 864-865.

Khan, J., Hardas, M., & Ma, Y., (2005), A Study of Problem Difficulty Evaluation for Semantic Network Ontology Based Intelligent Courseware Sharing, *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*.

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

Kunichika, H., Urushima, M., Hirashima, T., & Takeuchi, A., (2002), A Computational Method of Complexity of Questions on Contents of English Sentences and its Evaluation, *IEEE Proceedings of the International Conference on Computers in Education (ICCE'02).*

Li, T., & Sambasivam, S., (2003), Question Difficulty Assessment in Intelligent Tutor Systems for Computer Architecture, *Proc. ISECON 2003*, San Diego, vol. 4112.

Self, J., (1999), The defining characteristics of intelligent tutoring systems research: ITSs care, precisely, *International Journal of Artificial Intelligence in Education*, 10, 350-364.

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

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