Design and Implementation of an Affective ITS

María Lucía Barrón-Estrada, Ramón Zatarain-Cabada,
Rosalío Zatarain-Cabada, and Arminda Barrón-Estrada

Instituto Tecnológico de Culiacán, Culiacán Sinaloa,
Mexico

{rzatarain, lbarron}@itculiacan.edu.mx,
luciabarron@gmail.com

Abstract. We present the different steps used for developing an Affective and Intelligent Tutoring System for Mathematics Learning. The intelligent tutoring system evaluates cognitive and affective aspects of the users or students in order to present the learning material to them. The whole system applies a fuzzy logic system to decide the next exercise and a neural network to recognize user emotions.

Keywords. Intelligent tutoring systems, social networks, neural networks, emotion recognition.

1 Introduction

The work of a tutor is to teach and train a student through individualized instruction. This personalization that a human tutor achieves with the student is done through different ways of adapting the educational material to student needs. In order to do this job, the tutor makes use of his academic knowledge, his experience, and his observations of the student. On the other hand, an Intelligent Tutoring System (ITS) combines methods of Artificial Intelligence (AI) and education, with the aim of creating a flexible and interactive environment that considers the cognitive and affective states of students [1-3].

In this paper we present the methodology used to implement an affective and intelligent tutoring system for learning multiplication and division operations for third-grade students. The ITS incorporates a fuzzy logic system for assigning the next exercise to the student, artificial neural networks for recognizing student emotions, and Knowledge Space Theory [4] for structuring and representing the knowledge domain.

The paper is organized as follows. In Section 2 we present the design of the ITS. Section 3 gives information about its implementation. Section 4 comments some results and finally conclusions are presented in Section 5.

2 Design of the ITS

Before coding software, you must first make a sketch of software components that perform the functions for which they are intended. Here begins the design phase, where you get an overview of how a system works with the correct distribution of components based on your proposed software architecture. This will guide the implementation process.

2.1 ITS Architecture

In this section we present the ITS architecture (figure 1) which is formed by three main modules: The Expert Module called Domain Module, which represents the knowledge of the expert and is handled through different concepts related to Knowledge Space Theory. The knowledge base of this module is stored using a particular kind of XML format.

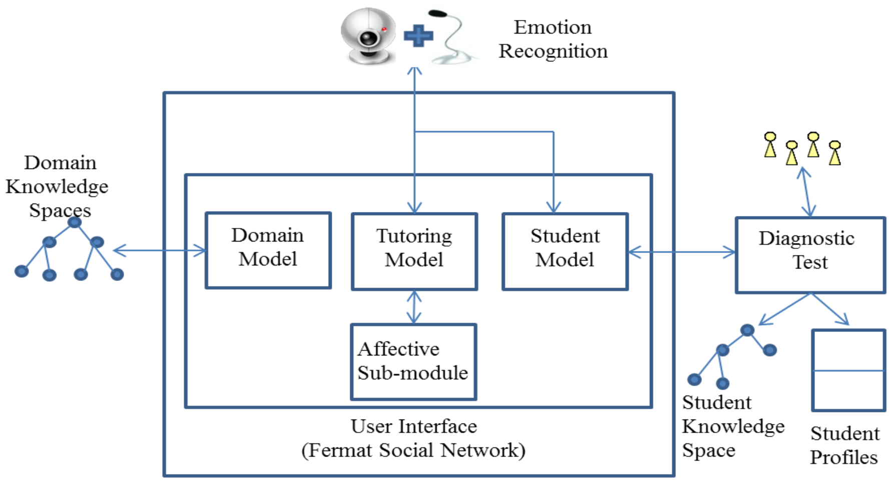

Fig. 1. ITS Architecture.

The Student Module provides the information about student competencies and learning capabilities through a diagnostic test. The student module can be seen as a sub-tree of all knowledge possessed by the expert and a student profile. For every student there is a static profile, which stores particular and academic information, and a dynamic profile, which stores information obtained from the navigation on the tutor and from the recognition of emotions.

The Tutoring Module presents the exercises to the students according to the level of the problem. We implemented production rules (procedural memory) and facts (declarative memory) via a set of XML rules. Furthermore, we developed a new knowledge tracking algorithm based on fuzzy logic, which is used to track student's cognitive states, applying the set of rules (XML and Fuzzy rules) to the set of facts.

2.2 Affective State Recognition

Affective State Recognition is based on Ekman's theory [5], which recognizes ten emotions, but we are only working with seven emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. We use Kohonen Neural networks [6] with 20X20 input neurons and 7 output neurons representing the emotion.

3 Implementation of the ITS

One of the main advantages of our ITS is that it works within a social network in a Web environment. Creating a web application has a large number of options for its construction. The tools and programming languages that were chosen for the development of intelligent tutoring system were:

- HTML5: For structuring and presenting the contents in the web.
- Java: For programming the ITS.
- JavaScript: For working with the user interface (the social network).
- JSP: (Java Server Pages): For dynamic creation of contents using Java. MySQL: For Data base Management.
- JSON (JavaScript Object Notation): For data interchange.

```
division([

        {"divisor":9,"dividendo":[1,0,8],"cociente":[0,1,2],"residuo":[1,0],"mul":[9,18]},
        {"divisor": 2,"dividendo":[4,2],"cociente":[2,1], "residuo":[0,0], "mul":[4,2]},
        {"divisor": 11,"dividendo":[1,0,0],"cociente":[0,0,9], "residuo":[1], "mul":[99]},
        {"divisor": 10,"dividendo":[5,0,0],"cociente":[0,5,0], "residuo":[0,0], "mul":[50,0]},
        {"divisor": 5,"dividendo":[7,2,5],"cociente":[1,4,5], "residuo":[2,2,0], "mul":[5,20,25]},
        {"divisor": 20,"dividendo":[1,1,2],"cociente":[0,0,5], "residuo":[019], "mul":[100]},
        {"divisor": 2,"dividendo":[4,0,9],"cociente":[2,0,4], "residuo":[0,0,0], "mul":[4,0,8]},
        {"divisor": 20,"dividendo":[5,0,2,0],"cociente":[0,2,5,1], "residuo":[10,2,0], "mul":[40,100,20]},
        {"divisor": 14,"dividendo":[1,3,2],"cociente":[0,0,9], "residuo":[6], "mul":[126]},


        ]);
```

Fig. 2. Structure of a JSON File for a Division.

3.1 The Expert Module

This module was created from JSON files. These contain all the information a student needs to know about how to solve a math division. The structure of these files is presented in Figure 2.

Figure 2 shows the basic structure of a JSON file. This case consists of an array of objects which contain the attributes of "divisor" and "dividend" that are shown to the student. Following are the attributes: "quotient", "remainder" and "mul", which contain the correct answers. There are other files, which are chosen according to the difficulty determined by the components.

3.2 The Student Module

It was implemented to determine the initial level of knowledge that a student possesses. For this, we implemented in HTML5 and JavaScript a diagnostic test whose answers are stored in a JSON file with a similar structure to that of Figure 2. The Student answers are compared with the contents of the JSON file. Each question has a value and a student score is determined by the following formula:

Student Score = Total Earned Points / Total points for all questions

The result is evaluated by a small algorithm to determine the difficulty in which the module tutor should start with the course.

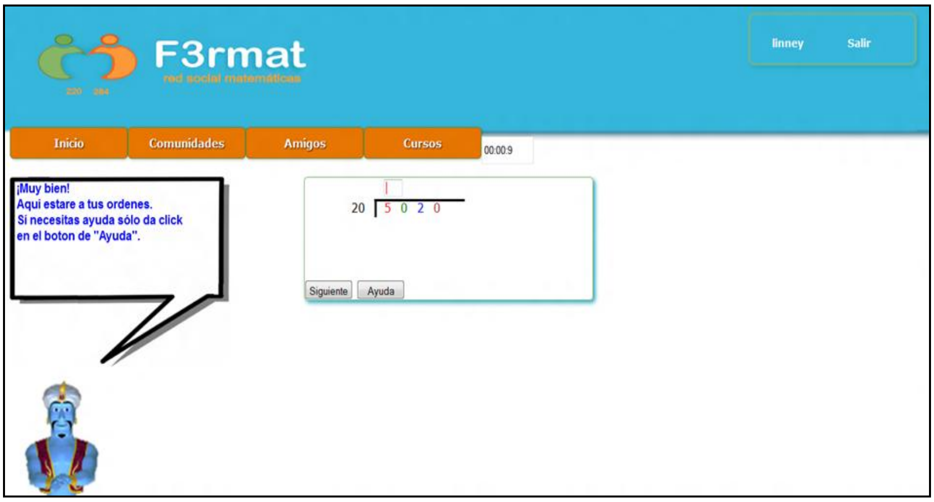

Fig. 3. Main Interface of the Intelligent Tutor.

3.3 The Tutor Module

This module is mainly based on the theory of cognition ACT-R [7]. In this part, the student solves exercises with the help and support of the intelligent tutor. Figure 3 shows the interface of this module.

Once inside the interface, students can enter answers they think are correct, while the tutor dynamically checks the corresponding JSON file. The initial difficulty is that which was determined in the student module. The difficulty of the next exercises can be modified depending on what the fuzzy expert system determines. The functionality of how responses are evaluated and the path taken by the process of solution is shown in Figure 4.

The premise is simple. The ITS waits for the entry of a value, and verifies that the value is correct to move to the next box and wait for the next value. If the answer is incorrect, the tutor sends a message through a pedagogical agent about the type of error. This is repeated until the division is complete. During this process the student can make use of two buttons located below the division operation. Button "Help" sends tips or advices to the student through the pedagogical agent. Button "next" moves to the next operation.

To determine the emotion, we first take the image which is transformed to a more basic form. Based on this picture we get the feature points that minimize the set of input data to the neural network. We use a Kohonen Neural network with 20X20 input neurons and 2 output ones representing the emotion. For the detection of emotions in the voice, this is captured primarily through the computer microphone and then is normalized. Then we apply the technique to characterize components analysis (PCA) to the signal representing the voice. After using the SFFS method [8] we obtain an optimal set of features that will feed the neural network. Each neural network used to recognize emotions produces an output. All outputs of each neural network are integrated using fuzzy logic which gives us a final result that is the emotion of the user that the system recognizes.

Table 1. Fuzzy Values for Variable Difficulty.

	Difficulty (%)	**Normalized Values**
Very Easy	0% - 10%	0 – 0.1
Easy	0% - 30%	0 – 0.3
Intermediate	20% - 80%	0.2 – 0.8
Difficult	70% - 100%	0.7 – 1.0
Very Difficult	90% - 100%	0.9 – 1.0

Fuzzy Expert System: The tutoring module also has an application made in java for reasoning with fuzzy logic. This program takes input fuzzy variables such as time, number of errors and number of helps. In a way the student's performance is reflected by such variables as he/she works with an exercise. We implemented a Fuzzy Expert System that eliminates arbitrary specifications of precise numbers and creates smarter decisions, taking into account a more human reasoning. Fuzzy sets are described in table 1 and figure 5 for linguistic variable Difficulty with fuzzy values very easy, easy, intermediate, difficult, and very difficult (“muy fácil”, “fácil”, “básico”, “difícil” and “muy difícil” in Spanish).

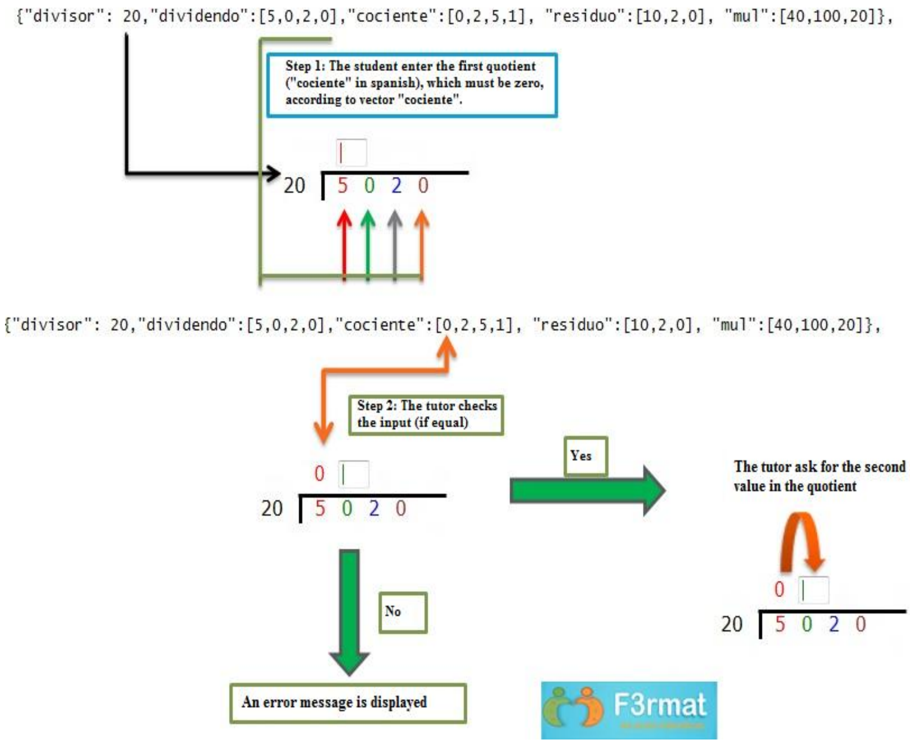

Fig. 4. Evaluation of a division.

On the other hand, some of the fuzzy rules which determine the degree of difficulty of the next student's problem are:

- If (Error is small) and (Assistance is small) and (Time is very fast) then (Difficulty is very difficult)
- If (Error is small) and (Assistance is normal) and (Time is slow) then (Difficulty is difficult)
- If (Error is big) and (Assistance is big) and (Time is very slow) then (Difficulty is very easy)

In this case the first fuzzy rule establishes that if a student (solving an exercise) had few errors, few assistances (helps), and finished in little time, the difficulty of the next exercise must be higher.

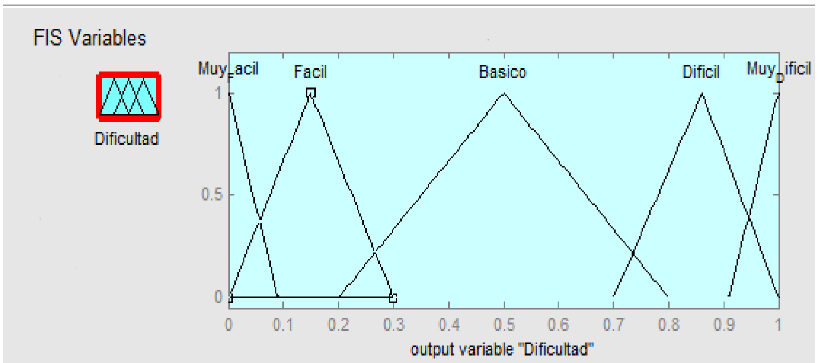

Fig. 5. Fuzzy sets for variable difficulty.

4 ITS Testing

The intelligent tutoring system was evaluated by a group of 72 children from third grade (Figure 6) in public and private schools. Before the evaluation we offered a small introduction of 15 minutes with the environment of the tool. We evaluated the subject of multiplications and divisions. We applied a test with different exercises before and after the students used the ITS. The results showed a good improvement in most students (more so in students with lower initial grades) using one of the two teaching methods for multiplication: traditional and lattice.

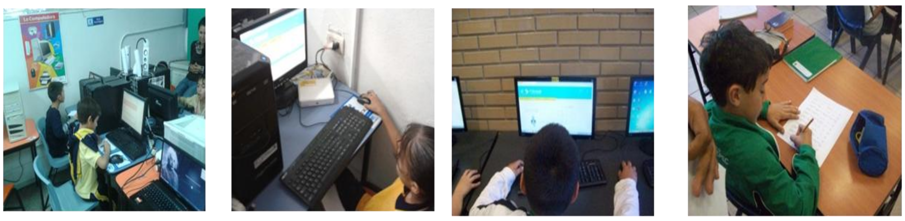

Fig. 6. Children testing the ITS in public and private schools.

5 Conclusions

Our intelligent tutoring system has been implemented for about one year. The results showed good improvement in the students. We are still working to adapt the application of emotion recognition with the intelligent tutoring system. One of the problem is that the ITS run in the web and the recognizer is still being tested in a desktop environment. We believe we will finish it in the next three months.

References

1. Arroyo, I., Woolf, B., Cooper, D., Burleson, W., Muldner, K., Christopherson, R.: Emotions sensors go to school. In: Diminitrova, V., Mizoguchi, R., Du Boulay, B., Graesser, A. (eds.) Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 17-24. IOS Press, Amsterdam (2009)
2. D'Mello, S.K., Picard, R.W., Graesser, A. C.: Towards an affective-sensitive AutoTutor. Special issue on Intelligent Educational Systems IEEE Intelligent Systems 22(4), 53-61 (2007)
3. Kort, B., Reilly, R., & Picard, R. W. An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy—Building a Learning Companion. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT '01). IEEE Computer Society, 43-50 (2001)
4. Doignon, J. –P. and Falmagne, J. C.: Knowledge Spaces. Springer-Verlag (1999)
5. Ekman P, Oster, H.: Facial expressions of emotion. Annual Review of Psychology 30:527-554 (1979)
6. Kohonen, T.: Self-Organization and Associative memory, Springer-Verlag, Third Edition, (1989)
7. Anderson, R., Boyle, C. F., Corbett, A. T., & Lewis, M. W.: Cognitive modeling and intelligent tutoring. Artificial Intelligence, 42, 17-49. Doi: 10.1016/0004-3702(90)90093-F. (1990)
8. Pudil P, Novovičová J, Kittler J.: Floating search methods in feature selection. Pattern Recognition Letters 15 (11) (1994)