Abstract— We present Fermat, an Intelligent Social Network for Mathematics Learning, which integrates an Intelligent Tutoring System as an extra feature to help students to improve the teaching and learning process. The intelligent tutor takes into account both cognitive and affective aspects. The social network and the affective tutoring systems are accessed from the web. Initial results in math show the benefits of using the tutoring system besides the all-known benefits of using a social network as the user interface.

Keywords: Intelligent Tutoring Systems, Affective Computing, Social Network, Student Affect

I. INTRODUCTION

Affect detection systems observe and study the face, speech, conversation and other human features to detect frustration, interest, boredom, etc. using hardware sensors like web or blue eyes cameras [1, 2], microphones [3], and conversational dialogues [4]. On the other hand, affect response systems find the best way to handle and improve a student's negative emotion. There are excellent research works related to this problem [4, 5].

In this paper we present an affective intelligent tutoring system embedded in a learning social network which is being used to improve poor results in ENLACE test (National Assessment of Academic Achievement in Schools in Mexico). ENLACE is the standardized evaluation of the National Educational System, applied to students in Grades 1-9 in public and private schools. The test measures learning in math, Spanish, and a third subject that changes every year.

II. FERMAT SYSTEM

The Learning Social Network Fermat has the basic functionalities in all social networks, but its main feature is that it includes an ITS that presents the course content in a personalized style.

Users of the network are associated with personal, academic and affective information, which is statically and dynamically stored in a profile. The static profile contains the initial information of the user (e.g. personal and academic information). The dynamic profile will be updated according to the user interaction with the network and the ITS, considering cognitive and emotional aspects.

Cognitive features are obtained according to the history and the results we get from user examinations. Emotional features are obtained by sensors that are monitoring the user's emotions.

III. FERMAT AFFECTIVE ITS

The intelligent tutoring system (ITS) for Fermat adopts the traditional trinity model known as the four-component architecture where a user interface has access to the three main modules: expert, student, and tutoring modules. Figure 1 shows the complete architecture of the Fermat ITS.

Expert Module: The knowledge acquisition and representation for the expert module is a major problem which we handle through different concepts related to Knowledge Space Theory [6]. This theory provides a sound foundation for structuring and representing the knowledge domain for personalized or intelligent tutoring. It applies concepts from combinatory theory and we use it to model particular or personalized tutors according to different cognitive styles.

A course in Fermat can be seen as a discipline-specific knowledge space (a particular tree diagram) containing chapters which in turn contain subjects. The whole tree represents the expert knowledge. The knowledge base of this module is stored using a particular kind of XML format.

Student Module: This module provides the information about student competencies and learning capabilities. Fermat realizes what the student's knowledge is through a diagnostic test. A Fermat student module can be seen as a sub-tree of all knowledge possessed by the expert and a student profile, as shown in the right part of figure 1. The representation is based on a model called "Overlay", where the student's knowledge is a subset of the expert knowledge.

As the student uses the intelligent tutor he expands this subset [7]. 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.

When a student first accesses the ITS, he has to answer the diagnostic test, which enables the construction of the student knowledge space. With the diagnostic test each question has a difficulty level, and depending on the level of...
difficulty, we give different weights to the answers. Difficult questions worth 3 points, intermediate questions worth 2 points and easy ones worth 1 point.

For the student's grade we use the following formula:

\[
\text{Student's Grade} = \frac{\text{Total Points scored}}{\text{Sum of points in the questions}}
\]

After obtaining the test results, a small algorithm determines the level of student learning and the teaching method. The algorithm used for computing and assigning the difficult level and the multiplication method do the following:

- If \( \text{Grade} < 4 \) Then Question level = easy; Method = lattice
- Else If \( \text{Grade} < 5 \) Then Question level = easy; Method = traditional
- Else If \( \text{Grade} < 9 \) Then Question level = normal; Method = traditional
- Else Question level = difficult; Method = traditional

For the student's knowledge representation we use two categories: Topics, which stores the history of subjects; and student experience, which stores the history of grades by subject.

**Tutoring Module:** The Fermat tutoring module is mainly based on ACT-R Theory of Cognition. These types of tutoring systems are also named Model-Tracing Tutors or cognitive tutors. We implemented production rules (procedural memory) and facts (declarative memory) via a set of XML rules. Furthermore, we developed a new knowledge tracing 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. The benefit of using fuzzy or vague rules is that they allow inferences even when the conditions are only partially satisfied.

The tutoring system reads the rules, and presents the exercises according to the level in the problem. The student cannot move to the next state (input) unless he solves correctly all the exercises. During this transition he can ask for help and in case of mistake, an error message is displayed to help discover what the correct answer is. Once the student completes the exercise, the student profile is updated with information on the type and difficulty of the problem or exercise, as well as the amount of mistakes, assistances, and the time it took to solve the problem. These variables (Difficulty, assistances, errors and time) will be required to determine the next problem the student will solve. For the tutoring module, we implemented a Fuzzy Expert System that eliminates arbitrary specifications of precise numbers and create smarter decisions, taking into account a more human reasoning. Fuzzy sets are described in table 1 and figure 2 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).

<table>
<thead>
<tr>
<th>Difficulty (%)</th>
<th>Normalized Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Easy 0% - 10%</td>
<td>0 – 0.1</td>
</tr>
<tr>
<td>Easy 0% - 30%</td>
<td>0 – 0.3</td>
</tr>
<tr>
<td>Intermediate 20% - 80%</td>
<td>0.2 – 0.8</td>
</tr>
<tr>
<td>Difficult 70% - 100%</td>
<td>0.7 – 1.0</td>
</tr>
<tr>
<td>Very Difficult 90% - 100%</td>
<td>0.9 – 1.0</td>
</tr>
</tbody>
</table>

**Affect Recognition and Handling:** We still are working with recognition of emotions in the face and voice. The method we use for the detection of visual emotions is based on Ekman's theory [8], which recognizes ten emotions. We are working with seven emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. We use Kohonen Neural networks with 20X20 input neurons and 2 output ones representing the emotion. Each neural network used to recognize emotions produces an output. All outputs of each
neural network are integrated using fuzzy logic which gives a final result that is the emotion of the user.

![Figure 3. Cognitive and affective feedback of Fermat ITS](image)

Figure 3 shows an interface of the Fermat Tutor with an affective agent represented by the Genie character of Microsoft agent. Some of the actions of the Genie are Congratulate, Confused, Explain and Suggest. These actions were the results of a study conducted to evaluate the expressivity of the animated agents [9]. In these surveys, 20 teachers were asked to select appropriate affective actions to be presented according to several tutorial scenarios. We want the agent helps students to learn and to foster a good affective state; hence we use decision theory to achieve the best balance between these two goals. The decision process is represented as a dynamic decision network (DDN). Our model uses multi-attribute utility theory to define the necessary utilities [10]. That is, the DDN establishes the tutoring action considering two utility measures: one on learning and one on affect, which are combined to obtain the global utility by a weighted linear combination. These utility functions are the means that allow educators adopting the system to express their preferences towards learning and affect.

IV. RESULTS AND DISCUSSIONS

Fermat social network along with its intelligent tutor was evaluated by a group of children from third grade (Elementary School). There were 25 children who tested the tool and the tutoring system. We evaluated the subject of multiplication. After the evaluation we applied a quiz to the group. In addition, we applied a questionnaire to the group using Fermat. With respect to the results, the graph in figure 4 shows the benefits of using Fermat. We can see from the results, the difference before and after using Fermat to solve multiplication exercises. The largest differences are seen in students with low initial level, to solve operations of multiplication.

V. CONCLUSIONS AND FUTURE WORK

The social network along with the intelligent tutor is being implemented using different software tools and programming languages. The presentation layer or part of the social network was implemented with CCS3, HTML 5 and Javascript. For the logical layer (mostly the intelligent tutor) we used Java and JSP. For the data layer we used XML and MySQL.

![Figure 4. Results of students using Fermat](image)

We need to do more tests with more students. We still are working with the emotion recognizer and we are adding more math operations to the ITS. However, the initial results are encouraging. We are implementing our own recognizers because we need to use them in a Web platform where the social network can be accessed from different computers like laptops or Mobile smartphones.

ACKNOWLEDGMENT

The work described in this paper is fully supported by a grant from the DGEST in México.

REFERENCES


