

Pattern Recognition for the Identification of Learning Styles on Educational Mobile and Social Network Tools

Ramón Zatarain Cabada¹, M. L. Barrón Estrada¹, and Carlos A. Reyes García²

¹ Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe, 80220, Culiacán Sinaloa, Mexico

{rzatarain, lbarron}@itculiacan.edu.mx

² Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, 72840, Puebla, Mexico
kargaxxi@inaoep.mx

Abstract. In this paper we present an implementation of pattern recognition techniques as the central part of an adaptive learning social network to be used as an authoring and learning tool. With this tool, adaptive courses, intelligent tutoring systems and lessons can be created, displayed and shared in collaborative and mobile environments by communities of instructors and learners. To show the maturation process to end up with the adaptive tool called Zamná, we first show the development of three previous intelligent tutoring systems with authoring and personalizing capabilities. In most of them the Felder-Silverman model is followed to tailor courses to the student's learning style. Several pattern recognition approaches are applied to identify the student's learning style. An introduction of a social learning network to create, view and manage adaptive intelligent tutoring systems, and some innovative methods to identify the student's learning style, are the contributions of this paper.

Keywords. Adaptive mobile learning, social learning networks, authoring tools, learning styles, artificial neural networks, fuzzy systems.

Reconocimiento de patrones para la identificación de estilos de aprendizaje en herramientas de educación móvil y en redes sociales

Resumen. En este trabajo se presenta la aplicación de las técnicas de reconocimiento de patrones en la parte central de una red de aprendizaje social adaptativa para ser utilizado como una herramienta de aprendizaje y de creación. Con esta herramienta, los cursos de adaptación, sistemas tutoriales inteligentes y los cursos se pueden crear, visualizar y compartir en entornos colaborativos y móviles de las comunidades de profesores y alumnos. Para mostrar el proceso de

maduración y para terminar con la herramienta adaptativa llamada Zamná en primer lugar mostramos el desarrollo de tres sistemas tutores inteligentes anteriores con capacidades de personalización. En la mayoría de ellos se utiliza el modelo de Felder-Silverman para elaborar los cursos a la medida para el estilo de aprendizaje de cada estudiante. Varios métodos de reconocimiento de patrones se aplican para identificar el estilo de aprendizaje de estudiantes. La introducción de una red de aprendizaje social para crear, ver y administrar los sistemas de tutoría inteligente adaptativo, y algunos métodos innovadores para identificar el estilo de aprendizaje del alumno, son las contribuciones de este trabajo.

Palabras clave. Aprendizaje adaptativo móvil, redes sociales de aprendizaje, herramientas de escritura, estilos de aprendizaje, redes neuronales artificiales, sistemas difusos.

1 Introduction

New technologies like Web 2.0 applications have appeared during the last few years. These novel technologies not only provide quality retrieval of information but allow us to implement and hold its control. Moreover, users can perform *Harnessing Collective Intelligence* [18, 12] by adding more value to the information as they make use of it. Several well-known Web 2.0 applications are **YouTube**, **Wikipedia**, **Yahoo! Answers**, and **Digg**. As a matter of fact, the distinguishing technologies of Web 2.0 are Social Network Sites. A study identifying social learning networks and analyzing seven e-learning 2.0 systems (EctoLearning, Edu 2.0, eLearning Community 2.0, LearnHub, LectureShare, Nfomedia, Tutorom (Eduslide)), was presented by Ivanova in [12].

According to Murray et al. [17], by 2003 over twenty authoring tools for Intelligent Tutoring Systems had been developed. Authoring tools are classified according to the type of a tutoring system they help to produce; for example the author tool SIMQUEST [14] produces “simulation-based learning” systems, IRIS [1] creates “multiple knowledge types” systems, and InterBook [4] generates Intelligent/adaptive Hypermedia. A common denominator in all the named tools is that they are designed only for authors without taking the learner into account.

With the idea of giving learners a more protagonist role, many efforts have been done recently to capture the way students learn in order to model their learning style [7, 11, 16, 25]. The Index of Learning Style Questionnaire (ILSQ) is used to calculate learning styles with the support of Bayesian Networks, Linear Temporal Logic, or Neuro-Fuzzy Systems.

On the other side, some authoring tools have been developed with the capability of implementing mobile applications; among them are MyLearning [2], Test Editor [20], or mediaBoard [3]. In these systems, PocketPC orientation, quiz editing or game-based learning are some of the approaches taken. All of them are authoring tools without the capability to be adapted to the student learning style, and cannot be interchanged among different operating system platforms.

Trying to overcome the particular limitations inherent to each kind of the described tools, we have developed an adaptive learning social network tool named Zamná. The tool was designed with most typical functionalities like new user’s registration, profiles and communities’ creation, etc, and with new functionalities like intelligent/adaptive learning in mobile devices. A new methodology to identify learning styles, based on Self-Organizing Maps is also introduced. This process is executed whenever a tutoring system is displayed, either in our system or in a mobile device. In general, our system is integrated with components coming from different domains like: Social Learning Networks, Mobile Learning, Adaptive or Web-based Systems, Artificial Neural Networks, and Intelligent Tutoring Systems.

This paper is organized as follows: Section 2 is devoted to explain Learning Styles Models. From Section 3 through Section 6 we present four different approaches to implement learning style classification. In Section 7 similar tools are shown. Finally, in Section 8 we present our conclusions and give some suggestions for future work.

2 Learning Style Models Feedback

Models of learning styles categorize both the ways in which students learn and how teachers teach. Their main objective is that in each category holding the model, the learning needs of students are met [9]. Taking into account the majority of existing models concerning learning styles, there are five major families [8]. Figure 1 shows four classes of models, which relate to theories of learning styles.

The model of learning styles of Gregorc was presented by Anthony Gregorc and Kathleen Butler. The objective of the model is to provide an organized structure of how the mind works. In this model there are four learning styles: Concrete-Sequential, Abstract-Random, Abstract-Sequential and Concrete-Random. Howard Gardner maintains the idea that every student is equipped with different types of intelligence. Gardner’s theory is known as Multiple Intelligences, he proposes that each individual has different intelligences at different levels. Gardner also states that every person has a unique cognitive profile. According to the multiple intelligence theory, there are nine basic types of intelligences: Visual-spatial, Verbal-linguistic, Logical-mathematical, Bodily-kinesthetic, Musical-rhythmic, Interpersonal, Intrapersonal, Naturalistic, and Existential.

The Myers-Briggs model is known as the Type Indicator Myers-Briggs (MBTI for short). The MBTI identifies four scales in which places all subjects. The scales are extraversion-introversion, sensitive-intuitive, thinking-feeling, and judgment-perception. From all the combinations of the scales, we obtain 16 types of personalities. Each of the combinations of types is described with four letters (one for each scale). The Felder-Silverman model was proposed by

Richard Felder and Linda Silverman in 1988 [9]. The model includes four dimensions or categories, two of which replicate features found in the models of Myers-Briggs and Kolb. The four dimensions in the model are related to perception (sensory-intuitive), processing (active-reflective),

input presentation (visual-verbal) and understanding (sequential-global). Learning styles are obtained by the combination of all categories. Thus, it is possible to have 16 different learning styles.

Gregorc	Gardner	Myers-Briggs	Felder-Silverman
<i>Concrete-Sequential</i> <i>Abstract-Random</i> <i>Abstract-Sequential</i>	<i>Verbal-linguistic,</i> <i>Logical-mathematical,</i> <i>Bodily-kinesthetic,</i> <i>Musical-rhythmic,</i> <i>Interpersonal,</i> <i>Intrapersonal,</i> <i>Naturalistic,</i> <i>And</i> <i>Existential</i>	<i>Extraversion-introversion,</i> <i>Sensitive-intuitive,</i> <i>Thinking - feeling,</i> <i>and</i> <i>Judgment-perception</i>	<i>Perception (sensory - intuitive),</i> <i>Processing (active - reflective),</i> <i>Presentation (visual - verbal) and</i> <i>Understanding (sequential-global)</i>

Fig. 1. Main learning style model

3 First Approach: Modeling Learning Styles with a Neuro-Fuzzy Network

The first approach to identify students' learning styles was done using a neuro-fuzzy network (NFN) which was included in **MLTutor**, an authoring tool which allows building personalized or intelligent tutoring systems to be used in both learning settings: distance and mobile learning. The tool can be employed for developing learning material with SCORM Learning Objects [21]. The learning material for mobile tools uses a neuro-fuzzy mechanism to identify and predict learning styles in order to provide an adapted learning scheme. The learning styles are based on Gardner's Pedagogical Model of Multiple Intelligences [10].

3.1 MLTutor Architecture

Figure 2 presents the architecture of MLTutor, in which it can be noticed that the tool has two main editors: the content editor and the fuzzy set editor. An author can create a tutoring system by first

building a course structure using the content editor. The author can introduce learning material by using a simple editor included in the tool. Other important learning materials that the authors can include into each of the study units are quizzes. The quiz is essential for the dynamic courseware generation because based on test results, neural networks classify learning styles. When the author introduces the learning material, four different instances should be created, according to Gardner's Pedagogical Model of Multiple Intelligences:

- Logical/Mathematical,
- Verbal/Linguistic,
- Visual/Spatial and
- Musical/Rhythmic.

There is a special interface in the content editor for helping the author when building this material.

The fuzzy set editor is designed to let the user to define Fuzzy Membership Functions for the Fuzzy Inputs, for which there are seven linguistic variables defined for a user. They are: answer selection order, correct answers, quiz spent time, topic spent time, tries until correct answer, visits to a question, and visits to a topic. Each of the

seven linguistic variables is described by three different values: low, average, and high. The Fuzzy inference process maps input linguistic values to output multiple-intelligence styles. The output of the fuzzy set editor can be m-learning material in XML format along with a predictive engine that employs a neuro-fuzzy inference algorithm [18]. It operates online using present and former information for each individual learner. At the start of each learning unit, predictions are made to establish the learners adequate learning material, based on former information built in the fuzzy set edition phase. Afterwards, the neuro-

fuzzy algorithm will learn from present information and will make adjustments, if necessary, to another best suited learning style. Another option to the output of the fuzzy editor is to export the learning material to SCORM format. The benefit of this format is the availability of the material in any distance learning environment. When a mobile course is exported to a mobile device, a XML interpreter is added to the course. This interpreter has the work of displaying the material of the course into the mobile device, according to the chosen kind of intelligence.

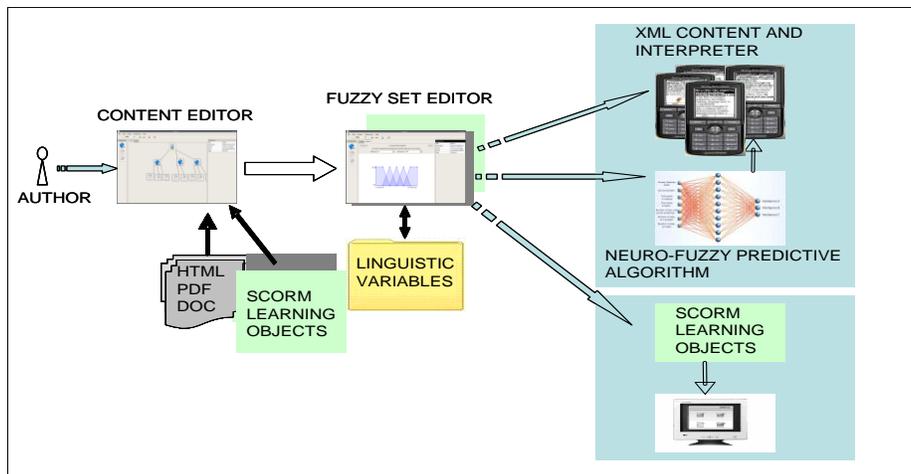


Fig. 2. MLTutor architecture

3.2 Neuro-Fuzzy Predictive Model

Figure 3 shows a reduced version of the MLTutor Neuro-Fuzzy system (just two linguistic variables are shown) implemented to determine types of intelligence from former and current data, and to make adjustment to new learning styles upon input data changes. The whole system is formed by seven linguistic variables, corresponding to four kinds of learning styles.

As we can observe in Figure 3, the system configuration consists of input and output layers, and three hidden layers that represent membership functions and fuzzy rules. The complete input layer has seven neurons representing our seven linguistic variables. The

input layer sends out external crisp values directly to the next layer without any processing. The output of this layer is as follows:

$$y_i = x_i, \quad (1)$$

where x_i and y_i are input and output respectively.

Each neuron of the input layer is connected to three neurons on layer 2 (fuzzyfication layer). In this layer the crisp values from the input layer are transformed to their corresponding linguistic fuzzy sets (low, average, or high sets). Every neuron of layer 2 represents a fuzzy set for each one of the seven linguistic variables (see Table 1). The output of every node in layer 2 is the membership degree of each input value to each fuzzy set. Every neuron of Layer 3 or fuzzy rule layer

represents a fuzzy rule (R1, R2, R3, etc.). Every fuzzy rule neuron takes inputs from layer 2 neurons. The rule is evaluated by the fuzzy intersection or t-norm, which in this case is the product operation. The output of each neuron is described as:

$$y_i = x_{1i} \times x_{2i} \times \dots \times x_{ki} \quad (2)$$

The output of layer 3 represents the weights of each one of the rules. The weights connecting layer 3 and layer 4 are changed or adjusted by training the neural network. The weight values are normalized through a series of iterations, by dividing each weight by the greatest weight found on each one of the adjustments or iterations and is represented by

$$w_{ni}(p + 1) = w_i(p) / w_{max}(p), \quad (3)$$

where w_{ni} is the normalized weight, p is the last iteration, w_i is the weight of connection i , and w_{max} is the greatest weight of the iteration. Layer 4 or output membership layer takes inputs from the fuzzy rule neurons and merges them by using fuzzy union, which in our case is the algebraic sum and is defined as

$$x_{1i} \oplus x_{2i} \oplus \dots \oplus x_{ji} \quad (4)$$

The outputs of layer 4 are the different learning styles (Logical/Mathematical, Verbal/Linguistic, Visual/Spatial and Musical/Rhythmic) produced in the fuzzy rule layer.

Layer 5 or defuzzification layer is the output of our system. In this layer the sum-product composition method is used [13]. The output of this last layer is a recommended learning style for the student. The learning algorithm we used is a modified back-propagation algorithm. Our algorithm takes a desired output (a default learning style), computed at the beginning of the iterations. Next, the algorithm computes the actual output and compares it with the desired output (dynamically adjusted according with the student test results). If there is a difference between the actual and the desired output, the error is calculated and propagated backwards through the network, updating or adjusting the weights between connected neurons. The neural network was trained also using Matlab (version 7.1). Then, the trained neural network was implemented using Java along with the XML interpreter.

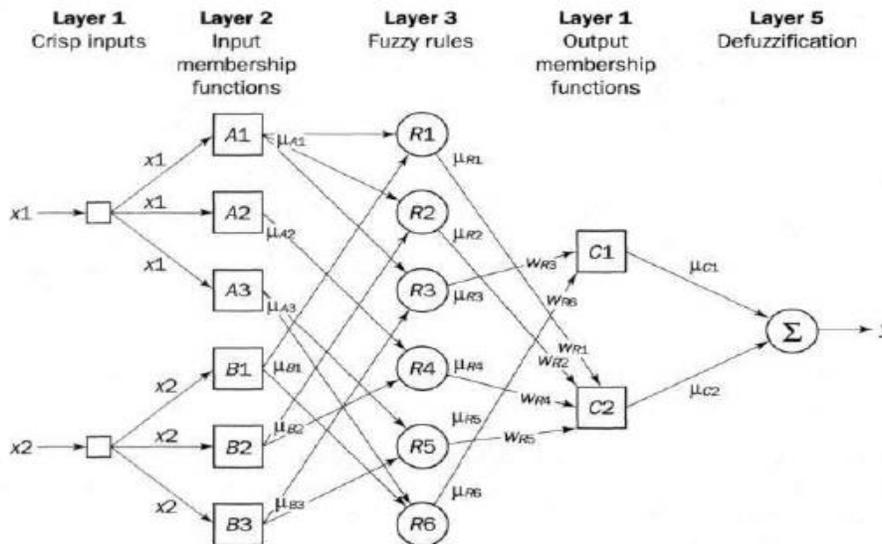


Fig. 3. Part of the MLTutor neuro-fuzzy system

Table 1. Fuzzy sets for two linguistic variables

TopicsSpentTime	CorrectAnswers
Range=[0 1000]	Range=[0 10]
NumMFs=5	NumMFs=5
MF1='Low':trapmf,[-225-25 100 300]	MF1='Low':trapmf,[-2.25 -0.25 1 3]
MF2='Medium-Low':trimf,[100 300 500]	MF2='Medium-Low':trimf,[1 3 5]
MF3='Medium':trimf,[300 500 700]	MF3='Medium':trimf,[3 5 7]
MF4='Medium-High':trimf,[500 700 900]	MF4='Medium-High':trimf,[5 7 9]
MF5='High':trapmf,[700 900 1025 1225]	MF5='High':trapmf,[7 9 10.2 12.2]

4 Second Approach: A NFN and a Genetic Algorithm

The second approach to identify students' learning styles was using a neuro-fuzzy network along with a Genetic Algorithm. In the following sections we present **EDUCA**, a Web 2.0 authoring tool which allow a community of authors and learners to create, share, and view learning materials and web resources in an adaptive environment which combines collaborative, mobile and e-learning methods. EDUCA applies different computational intelligence techniques like a neural network and a genetic algorithm for selecting the best learning style or a recommendation-web mining system for adding and searching new learning resources.

4.1 EDUCA Architecture

Figure 4 illustrates the overall architecture of EDUCA. As we can observe, there are two kinds of author: the main tutor (a teacher or instructor) and the community of learners. The student or learner is an important author of the course and participates actively adding learning resources to the courses. The learner has a user profile with information like GPA (the grade point average of the student), particular learning style, or recommended resources to the course. When the authors add learning material, they first create four different instances corresponding to four different learning styles according to Felder-Silverman Learner Style Model [9]. We implemented a fuzzy-neural network using the fuzzy input values previously defined. The output of the network is the learning style for each student taking a course. This model was complemented with the implementation of a

genetic algorithm for the optimization of the weights of the network. For our experiment purposes, the network was trained for 800 generations, using a population of 150 chromosomes. Besides, we created three sets of courses for high school students. Each course was elaborated in four different teaching styles according to the Felder-Silverman model. When a course is exported to a mobile device, a XML interpreter is added to the course. A SCORM file for the course can also be exported.

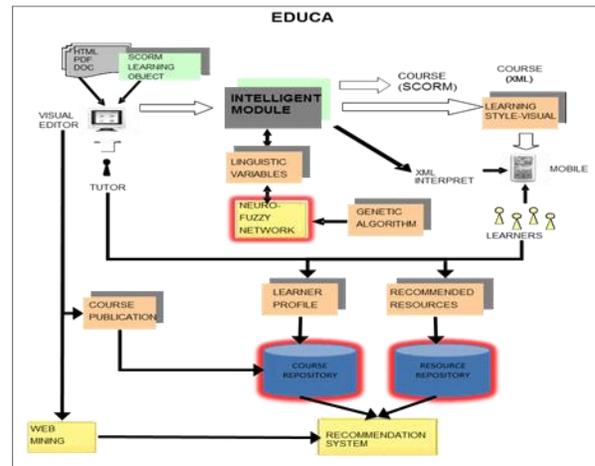


Fig. 4. EDUCA architecture

Once a course is created, a Course Publication Module saves it into a Course Repository. Whenever a learner accesses a course, a recommender system implemented in EDUCA presents links or Web sites with learning material related to the current topic. Such material is stored in a resource repository of EDUCA, which was searched previously by using Web mining techniques implemented also in EDUCA.

4.2 The Intelligent Module

The Intelligent Module takes as an input the learning material for the four different learning styles. Then, it creates a NFN used to classify the learning style of the user/student and produces as an output, an adaptive course (a special type of Intelligent Tutoring System). An adaptive course is structured with two components: A XML file, containing the learning material, and the XML Interpreter. The Interpreter uses the NFN as a dynamic classifier to show the learning material according to the best learning style of the user.

The first layer of the NFN has 7 neurons (see Figure 3) representing the 7 linguistic variables used for the identification of the learning style. Every neuron of layer 1 is connected to 3 neurons of layer 2 (fuzzyfication layer). Due to the fact that we are using triangular membership functions, the activation function for the layer 2 neurons is as follows:

$$\begin{aligned}
 y_i^{(1)} &= 0, \text{ if } x_i^{(1)} \leq a - b / 2 \\
 y_i^{(1)} &= 1 - 2 |x_i^{(1)} - a| / b, \\
 &\text{if } a - b / 2 < x_i^{(1)} < a + b / 2 \\
 y_i^{(1)} &= 0, \text{ if } x_i^{(1)} \geq a + b / 2
 \end{aligned}
 \tag{5}$$

where a and b are the centre and width of the triangle, $x_i^{(1)}$ and $y_i^{(1)}$ are input and output of neuron i respectively.

The output of layer 3 represents the strength of each one of the fuzzy rules. The best weight values between layer 3 and layer 4 are calculated using a genetic algorithm. Layer 5 is the output of the NFN. We applied the *Centroid* technique to defuzzyfy. The value of the output is the learning style for the current student of the course.

4.2.1 Training the NFN with a Genetic Algorithm

At the beginning, we create a chromosome population with random values. Each element of the population represents the weights in the NFN (see Figure 5).

For encoding the chromosome, the weights of the NFN are sorted using a Bucket Sort Algorithm, ordering first by layer, and then by

neurons. The output of this algorithm is a chromosome. Each gene of the chromosome represents a weighted link in the NFN.

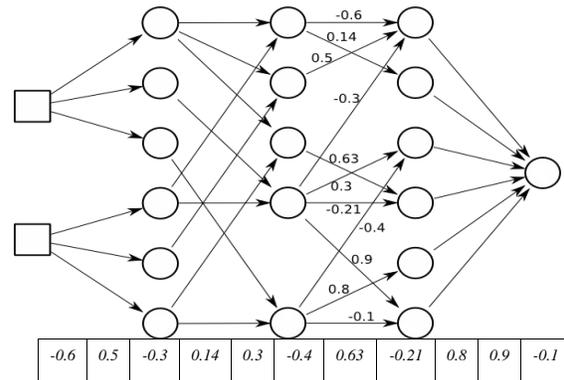


Fig. 5. Decoding the weights in the NFN into a chromosome

To evaluate the chromosome's performance, we assign each weight (gene) contained in the chromosome to the links of a respective NFN. Then we test the network with a set of training values. Last, the sum of squared errors is evaluated, which will be used as the fitness value of the tested chromosome.

In order to train the network, we create three sets of courses for high school students. Each course was presented in eight different styles for the Felder-Silverman model. The chosen courses were Teaching Digital Photography, Eolic Energy, and Introduction to Computers. In order to get their best learning style we applied the ILSQ test [9,22] to every student. We also designed a test to evaluate the course taken by the student. The input data to the network was the performance of the student under each course and the learning style of the randomly taken course. The Desired Output was the learning style calculated from Felder-Silverman Questionnaire. We made tests with two groups of 40 students.

4.3 Using the NFN on Cell Phones

Whenever a course is created and exported to a cell phone or PDA (using XML format), the NFN is

also exported to those devices, along with a XML parser or interpreter. This program runs any course stored in the mobile by reading the XML file where the course is stored. At the beginning, a randomly chosen learning style is assigned to the student or user of the course. Then, depending of the results from quizzes inserted into the learning material, the learning material is adapted to the best learning style (Visual, Verbal, Sequential, etc.) of the student.

An important feature is the possibility of tracing the different learning styles that a user or student maintains during lesson learning. The interpreter of the course optionally displays the current student learning style plus the values of fuzzy variables. This information is relevant for doing different types of analysis with respect to behaviors of the students and the way they learn. Figure 6 shows two snapshots of a handheld device (a cell phone) showing values of each learning style in a range between -100 to +100 (left device) and a dynamic sequence of the style behavior during the execution of the course (right device). Figure 7 presents the different steps to create a course (intelligent tutoring system) with

EDUCA. Following a clockwise order, first we create a tree structure for the course. Next, we add learning material using learning styles. In the third step we add recommended resources to the course. In the last step, the course is exported to the mobile device. When a user navigates in the course he can also rate this course. This option is part of the Web 2.0 functionality added to EDUCA.



Fig. 6. Dynamic sequencing of learning material

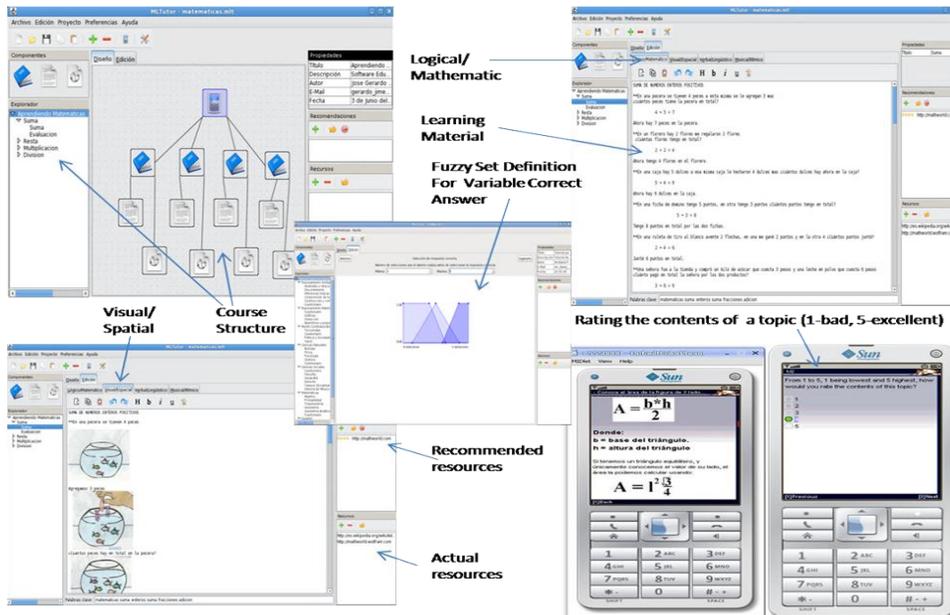


Fig. 7. Creation of learning material for a basic math course

5 Third Approach: A Self-Organizing Map

In this section, we present a new version of the authoring tool **EDUCA** using a self-organizing map (SOM) or Kohonen network [Kohonen, 89] to identify students' learning styles. Advantages of Kohonen networks include implementation simplicity, execution speed and a shorter training process; however maybe the most important advantage of these neural networks is that they do not require directions to perform training. During a training session, the SOM receives a number of different input patterns (the student learning style obtained from ILSQ test, the course learning style, and the student's grade in the course), discovers significant features in these patterns (Felder-Silverman learning styles) and learns how to classify the input.

5.1 A New EDUCA Architecture

Figure 8 presents the new architecture of **EDUCA**. The most important modification in this new version is a SOM which substitutes the NFN used in the first version. The functionalities in this new authoring tool are very similar for the user

with only a new editor for the creation of learning material. After the author creates a course or intelligent tutoring system, she/he can save it and export it to a Mobile Learning format used to display tutoring systems on mobile devices. The saved/exported file will enclose three elements: a XML file corresponding to the learning style model, a predictive engine for navigation purposes, and the Kohonen Neural Network for learning style classification.

Once a Tutoring System has been created, the module *Published Course* stores it in a Course Repository. We know that learners usually read tutoring systems stored in some kind of repository, but they also consult other learning resources in different web sites. For example, a student who needs to implement a LR parser in a compiler course, could use the Tutoring System stored in the course repository, but she/he could also consult extra resources in the Web. This material found by the student would be rated and recommended to be added to the regular compiler course. EDUCA uses a Hybrid Recommendation System [5] which stores new resources into the Resource Repository. Last, as a complement, EDUCA uses a Data or Text Mining Subsystem used to search resources in the Web.

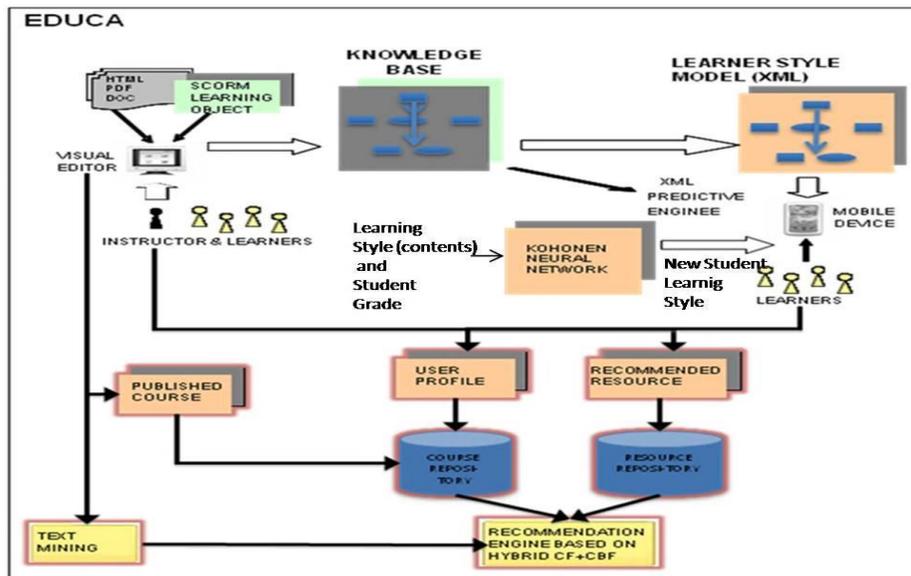


Fig. 8. New architecture of EDUCA

5.2 Training the SOM

An input vector in our neural network is described as:

$$D = [d_{fs} \ d_a \ p] \tag{6}$$

where D is the input vector, and it is formed by three elements: d_{fs} is the student learning style identified by applying ILSQ test; d_a is the learning style used in the learning material read by the student; and p is the student grade obtained in the test when the student had learning style d_{fs} and the course was offered using learning style d_a .

Vectors d_{fs} and d_a are also composed as:

$$d_a = d_{fs} = [c_1 \ c_2 \ c_3] \tag{7}$$

where c_1 c_2 c_3 represent three scores for Perception, Input, and Understanding Dimensions in the ILSQ test. Figure 9 shows part of the Kohonen network. We can see that the input layer has seven neurons, corresponding to the three-dimension vectors d_{fs} and d_a , plus the student grade p . The input data vector d_{fs} , vary between -11 to +11, that is massaged to the range -1 to +1 with the equation:

$$Massaged\ value_{11} = v_i / |v_{imax}| \tag{8}$$

where v_i is an ILSQ score, and v_{imax} is the maximum ILSQ score value (+11). As we observe in Figure 9, -3, an ILSQ score for visual/verbal learning style, is mapped to -0.272. The eight different learning styles we implement are for visual-verbal, sequential-global and sensitive-intuitive dimensions. To represent the three values of each dimension in vector d_a , we do not need to normalize the three values of the vector. A value of -1 represents the left side and a value of 1 represents the right side of each pair. This means that -1, -1, -1 represents the learning style *visual-sequential-sensitive*. On the other hand, the student grade, which varies between 0 and 100, is also massaged to the range -1 to +1, with equation:

$$vnj = 2vj / vjmax - 1 \tag{9}$$

where v_j is the student grade or score, and v_{jmax} is the maximum student grade (+100). For example, 70 is mapped to 0.4.

The Kohonen layer has 1600 neurons arranged in a structure of a 40x40 hexagonal grid. The output of the network consists of three signals or values ranging between -1 to +1. These data are then decoded to the ILSQ range (-11 to +11); they represent the learning style of the student.

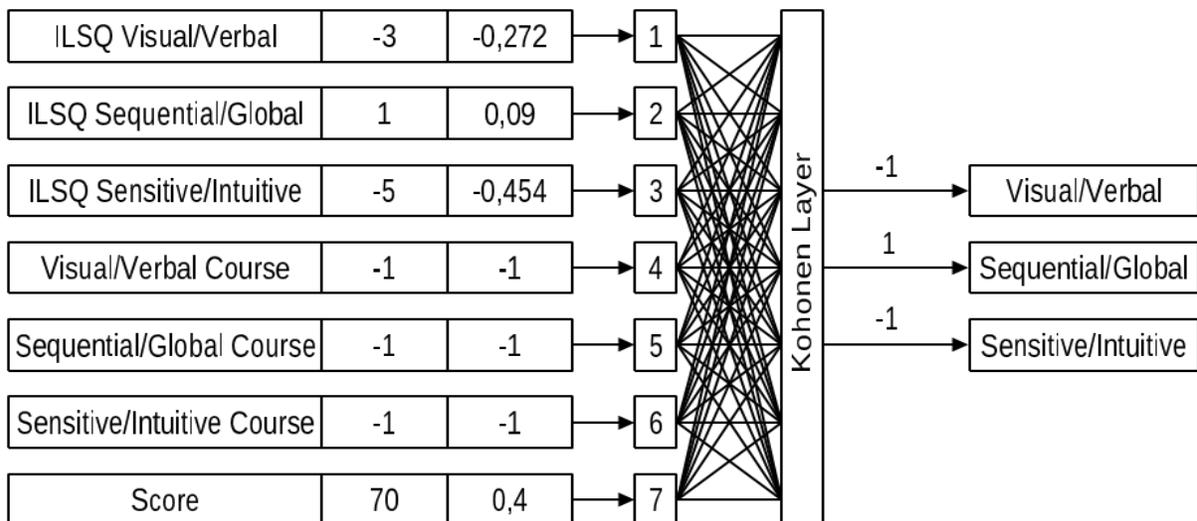


Fig. 9. The Kohonen network for learning styles

The procedure to train the neural network consists of three steps. The samples for this experiment were obtained from 140 high-school students. During the first step we applied to them the ILSQ test in order to identify the learning style of each student. In the second step we created three different introductory courses: Computer Science, Photography, and Eolithic Energy. For each course we produced eight versions, one for each learning style combination (visual/verbal, sequential/global, sensing/intuitive). Randomly, the students received each of the three courses, in only one learning style. We recorded the assigned learning styles of each student in the three courses. In the third step and after the students read the courses, they took their corresponding tests in order to evaluate their performance in each course.

With these three elements (student learning styles, course learning styles, and student evaluation) we created 140 input vectors for training the network. We trained the network with 6000 iterations, an initial neighbourhood ratio of 50% of the lattice size and a learning rate of 0.1. The network had 1600 neurons arranged in the form of a two-dimensional lattice with 40 rows and 40 columns, and every input vector had seven elements, as we established previously.

5.3 Using the Kohonen Network

As we stated before, the network is devoted to identify learning styles of students. The network is then exported along with a corresponding tutoring system and the predictive engine to the mobile (see Figure 8). When the student reads a tutoring system in a time t , the learning material shown to the students uses learning style g . If the student answers a quiz as part of the learning contents, then he receives a grade k .

Therefore, the network r takes as input the learning style of the displayed contents, and performs a search to find the winner-takes-all (best-matching) neuron j_x at iteration p , using the minimum-distance Euclidean criterion:

$$j_x(p) = \min \|X - W_j(p)\| = [\sum(x_i - w_{ij})^2]^{1/2} \quad (10)$$

where X is the input pattern vector and W is the weight vector.

After finding neuron j_x , the output of the network is a new student learning style, defined as

$$g(t + 1) = r(g(t), k) \quad (11)$$

where g is the student learning style, t is a time, r is the neural network, and k is the student grade.

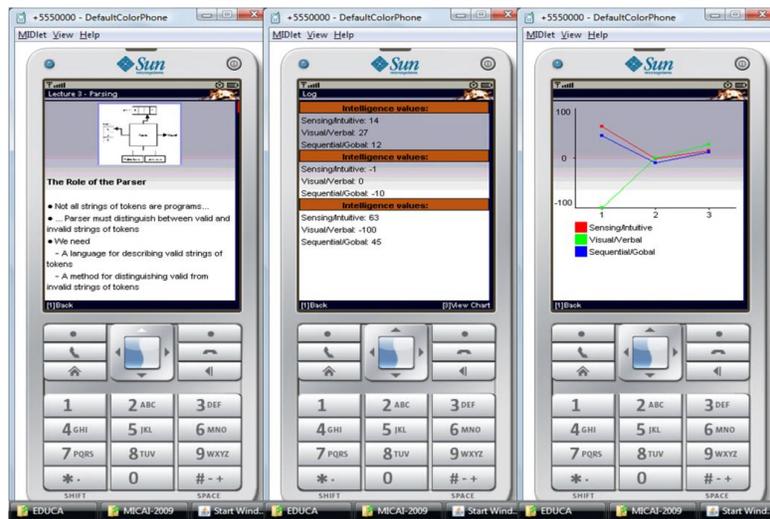


Fig. 10. A sample of the compilers course in a cell phone

This process of identifying the student learning style is performed every time the student answers a quiz on the tutoring system.

5.4 Tests and Results

The tool was used by a group of different users (authors), which consisted mainly of university professors/students. They produced different intelligent tutoring systems like Compiler Constructions, Introduction to Computer Science, teaching the Maya Language, and an introduction to Elementary mathematics. The intelligent systems were initially created by a professor and later the students were using and adding new learning resources with the recommendation system. Figure 10 presents three snapshots of the Compiler course running in a mobile phone (Sun Java Wireless Toolkit 2.5.2). The first mobile phone (left) shows learning material about the parsing topic. The second and third mobile phones give us a trace of the students' learning styles (in three stages) along the course.

6 A New Approach: Learning Environments as Social Networks

In Figure 11, we describe the architecture of the Web 2.0 tool *Zamná* that is a Social Network oriented to be used in education. As can be noticed from the diagram, users have access to *Zamná* through any browser. For any access, the user workspace will compose several components, which are; a news section, a user profile, an inbox part, a section of courses, communities, documents, lessons and friends. To create the component *profile* *Zamná* makes use of the intelligent module to identify the user's learning style. The intelligent module makes use of a *course viewer* for displaying the contents of a course according to the learning style identified by the intelligent module. The course materials are stored in data bases or repositories from where they are downloaded to be exported and studied on any mobile device. Another component is *Communities*, a set of networks, each one focused on a particular area of knowledge for specific purposes. The

communities are also stored in a dedicated repository. The same description holds for the component *Lessons*.

6.1 The Predictive Engine

The predictive engine is designed to dynamically identify the student's learning style every time a tutoring system is run. At any time, a new learning style can be evaluated as a result of the evaluations applied to the student.

Based on the nature of our problem, and on the experience acquired from the results of the application of the previous learning systems, we decide to use again an intelligent module to identify the user's learning style, which should be provided of unsupervised training abilities, good proved performance, and fast training potential. On those grounds, we based the intelligent module on the Kohonen's Self-Organizing Maps (SOM). The identification of the student's learning styles is a pedagogical discipline, which requires the direct observation and supervision of an expert pedagogue, so, we have thought that the SOM will automatically fulfill the role of an always available pedagogue performing such a task.

6.2 Tests and Results

After the SOM has been trained, we applied a validation process, for which we use seven parameter vectors, where four parameters are used to identify learning styles (three items to define the style of learning of the material provided to the student, and one which represents the student performance with that teaching material). Another three parameters contain the outcome response from the neural network. Table 2 shows some values for the input vectors with different parameters, along with expected values for each vector, used during training, and obtained results, which are the actual output of the network for each input vector.

The order of scales in the input vector is as follows: the first element corresponds to the scale *Visual/Verbal*, the second to the scale *Sensitive/Intuitive* and the third to the scale

Sequential/Global. Therefore, if one takes for example the first input vector shown in Table 2, we have a learning style configuration *Visual-Intuitive-Sequential*(-1 +1 -1). The output vectors are identified as *expected result* and *obtained result* and have the same order as the

input vectors. Similarly, when selecting the first vector of the set of expected results, (-0.8181 - 0.2727 -0.0909), is possible to know the learning style that the neural network report as a result (Visual-Sensitive-Sequential).

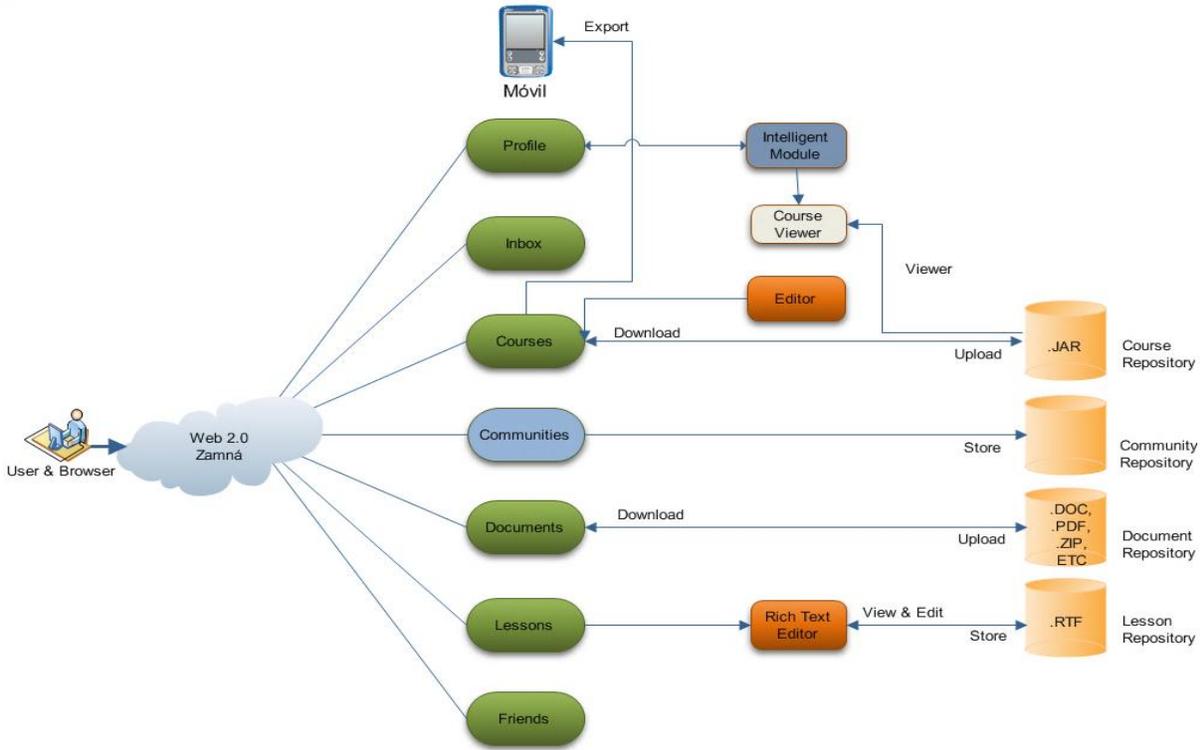


Fig. 11. Zamná architecture

Table 2. Test results in the training process of the neural network

INPUT VECTORS			EXPECTED RESULTS			OBTAINED RESULTS			
-1	+1	-1	+0.6	-.8181	-.2727	-.0909	-.6315	-.1447	-.3863
-1	-1	-1	+0.8	-1.000	+.0909	+.2727	-.4896	-.1071	-.0176
-1	+1	+1	+0.4	-.4545	-.0909	-.2727	-.0919	+.1137	-.4328
+1	+1	-1	+0.4	-.8181	+.8181	+.2727	-.3933	-.0844	-.4320
+1	+1	+1	+0.3	+.0909	-.0909	-.4545	-.2765	-.0365	-.4807
+1	-1	-1	+0.3	-.8181	-.0909	-.0909	-.7461	+.0429	+.2300
-1	-1	+1	+0.2	-.4545	+.0909	-.2727	-.7392	-.1687	-.3382
-1	+1	+1	+0.4	-.4545	-.0909	-.2727	-.0919	+.1137	-.4328
+1	+1	+1	+0.3	-.0909	+.2727	+.0909	-.3956	-.4314	+.4301
+1	+1	-1	+0.5	-.2727	+.2727	-.6363	-.5336	+.1870	-.3271
+1	-1	+1	0.0	-.2727	+.4545	-.2727	-.3730	+.0481	+.2212
-1	+1	+1	+0.8	+.0909	+.0909	-.0909	-.3133	-.0021	-.3617

7 Similar Tools and their Outcomes

The authoring tool SIMQUEST [14] produces “simulation-based learning” systems, IRIS [1] creates “multiple knowledge types” systems, and InterBook [4] generates Intelligent/adaptive Hypermedia. One shared feature of all of those tools is the separate role of authors and learners. For instance, in SIMQUEST one author creates a simulation model, learner interface, instructional design, and environment. In this case, the learner is never involved. With IRIS, the author produces an intelligent tutoring system by following two main phases: Specifying the general requirements of the tutoring system, and filling the learning contents of the ITS. Building adaptive educational material is a similar procedure: the developer has to create content objects and specify the links between them. Authoring tools for developing adaptive educational/tutoring systems like InterBook, SIGUE [6], NetCoach [24], and Ale [23] differentiate from each other only by the interface tools the authors operate. They use special markup languages or GUI interfaces.

8 Conclusions and Future Work

We have presented several pattern recognition techniques implemented as fundamental components of adaptive intelligent tools developed for education. The adapting capabilities were provided in different models, all of them oriented to the identification, either, of the type of intelligence or the learning style of the student. The development process to get to the present version of our social network learning tool is represented by the progressive and by themselves successful systems previously developed and tested.

MLTutor, an authoring tool which allows building personalized or intelligent tutoring courses to be used in both learning settings: mobile and e-learning. The learning material for mobile tools uses a neuro-fuzzy mechanism to identify and to predict learning styles in order to provide an adapted learning scheme. The learning styles are based on Gardner's Pedagogical Model of Multiple Intelligences.

In the second approach to identify students' learning styles the neuro-fuzzy network was complemented with a Genetic Algorithm for training. The resulting model was called EDUCA, a Web 2.0 software tool.

In the following version of the authoring tool EDUCA a self-organizing map was implemented to identify students' learning styles. The SOM added simplicity, execution speed, and a shorter and unsupervised training process to EDUCA. On the other hand, if we analyze the quality of results in Table 2 as hard numbers, which only indicate the preferred size for each scale, it is observed that 16.66% of the results obtained are consistent with the expected results on all scales, and 66.66% in at least two of the three scales. However, when the vectors are transferred to the interpretation of Felder and Soloman, and the numerical values are replaced with adjectives related to the preference of each dimension (strong with a difference between -1 to -6.66, weak with a difference between -6.66 to -3.33 and almost nonexistent with a difference between -3.33 to 0), the analysis show that the quality of results is very satisfactory.

Based on the previous experiences we developed the Web 2.0 tool Zamná, a Social Network to be used in education. It also uses a SOM to identify the user's learning style. Zamná was tested as both, an authoring tool, by collecting material for several courses in different versions for each learning style, and as learning instrument, by offering the courses to students, in the version corresponding to their already identified learning style. At present, our system continues being tested, evaluated and improved. We plan to further test the site with more groups of students from diverse levels and different areas of study. In addition, we want to test the courses created for mobile devices by following their impact and results of studying actual courses. In this case, we plan to first analyze the forms to provide the learning material that is most suitable to be managed in these mobile devices.

References

1. **Arruarte, A., Fernández, I., Ferrero, B. & Greer, J. (1997).** The IRIS Shell: How to build ITSs from

- Pedagogical and Design Requisites. *International Journal of Artificial Intelligence in Education*, 8(3), 341–381.
2. **Attewell, J. (2005a).** From Research and Development to Mobile Learning: Tools for Education and Training Providers and their Learners. *4th World Conference on mLearning*, Cape Town, South Africa, 1–6, Retrieved from <http://www.mlearn.org.za/CD/papers/Attewell.pdf>
 3. **Attewell, J. (2005b).** *Mobile technologies and learning*. Retrieved from <http://www.m-learning.org/docs/The%20m-learning%20project%20-%20technology%20update%20and%20project%20summary.pdf>.
 4. **Brusilovsky, P., Eklund, J. & Schwarz, E. (1998).** Web-based education for all: A tool for development adaptive courseware. *Computer Networks and ISDN Systems*, 30(1–7), 291–300.
 5. **Burke, R. (2002).** Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interactions*, 12(4), 331–370.
 6. **Carmona, C., Bueno, D., Guzmán, E. & Conejo, R. (2002).** SIGUE: Making Web Courses Adaptive. *Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems AH'02*, Malaga, Spain, 376–379.
 7. **Carmona, C., Castillo, G. & Millán, E. (2008).** Designing a Dynamic Bayesian Network for Modeling Student's Learning Styles. *Eighth IEEE International Conference on Advanced learning Technologies ICALT'08*, Santander, Spain, 346–350.
 8. **Coffield, F., Moseley, D., Hall, E. & Ecclestone, K. (2004).** *Learning Styles and Pedagogy in post-16 Learning: A Systematic and Critical Review*. England: Learning and Skills Research Centre. Retrieved from: www.hull.ac.uk/php/edskas/learning%20styles.pdf
 9. **Felder, R.M. (1996).** Matters of style. *ASEE Prism*, 6(4), 18–23.
 10. **Gardner, H. (1983).** *Frames of Mind: The theory of multiple intelligences*. New York: Basic Books.
 11. **Graf, S., Kinshuk & Liu, T. (2008).** Identifying Learning Styles in Learning Management Systems by Using Indications from Students' behavior. *Eighth IEEE International Conference on Advanced learning Technologies ICALT'08*, Santander, Spain, 482–486.
 12. **Ivanova, M. (2008).** Knowledge Building and Competence Development in eLearning 2.0 Systems. *Fifth International Conference on Knowledge Management*, Columbus, Ohio, USA, 84–91.
 13. **Jang, J.S.R. (1993).** ANFIS: Adaptive Network-based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665–685.
 14. **Jong, T., de Limbach, R., Gellevij, M., Kuyper, M., Pieters, J. & Joolingen, W.R. (1999).** Cognitive tools to support the instructional design of simulation-based discovery learning environment: the SIMQUEST authoring system. In van den Akker, J., Branch R.M., Gustafson, K, Nieveen, N. and Plomp, Tj., (eds.), *Design Approaches and Tools in Education and Training* (215–224). The Netherlands: Kluwer Academic Publishers.
 15. **Kohonen, T. (1989).** *Self-Organization and Associative Memory* (3rded.). Berlin: Springer-Verlag.
 16. **Limongelli, C., Sciarrone, F. & Vaste, J. (2008).** LS-PLAN: An Effective Combination of Dynamic Courseware Generation and Learning Styles in Web-based Education. 5th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems AH'08. *Lecture Notes in Computer Science*, 5149, 133–142.
 17. **Murray, T., Blessing, S. & Ainsworth, S. (2003).** *Authoring Tools for Advanced Technology Learning Environments: toward cost-effective adaptive, interactive, and intelligent educational software*. Dordrecht: Kluwer Academic Publishers.
 18. **Negnevitsky, M. (2005).** *Artificial Intelligence: A guide to Intelligent Systems* Harlow. England; New York: Addison-Wesley.
 19. **O'Reilly, T. (2005).** What is Web 2.0. Retrieved from <http://oreilly.com/web2/archive/what-is-web-20.html>
 20. **Romero, C., Ventura, S., Hervás, C. & De Bra, P. (2006).** An Authoring Tool for Building Both Mobile Adaptable Tests and Web-Based Adaptive or Classic Tests. *4th International conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'06. Lecture Notes in Computer Science*, 4018, 203–212.
 21. SCORM (The Sharable Content Object Reference Model) (s.f.). Retrieved from <http://www.adlnet.gov/capabilities/scorm>.
 22. **Soloman, B. A. & Felder, R. M. (1997).** Index of Learning Styles Questionnaire. Retrieved from <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>
 23. **Specht, M., Kravcik, M., Klemke, R., Pesin, L. & Hüttenhain, R. (2002).** Adaptive Learning Environment (ALE) for Teaching and Learning in WINDS. *Second International Conference on Adaptive Hypermedia and Adaptive Web-based*

Systems. AH'2002. Lecture Notes in Computer Science, 2347, 572–575.

24. Weber, G., Kuhl, H.C. & Weibelzahl, S. (2001).

Developing adaptive internet based courses with the authoring system NetCoach. Retrieved from wwwis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf.

25. Zatarain-Cabada, R., Barrón-Estrada, M.L., Sandoval, G., Osorio, M., Urías, E. & Reyes-García, C.A. (2008).

Authoring Neuro-fuzzy Tutoring Systems for M and E-Learning. *MICAI 2008: Advances in Artificial Intelligence. Lecture Notes in Computer Science*, 5317, 789–796.

He is the former President of the Executive Board of the Mexican Society for Artificial Intelligence (SMIA). His research interest areas are: Intelligent Systems, Fuzzy Sets and Logic, Fuzzy Neural Networks, Genetic Algorithms, Machine Learning, Knowledge Representation, Automatic Speech and Speaker Recognition, Infant Cry Recognition and General Pattern Classification.

Article received on 12/01/2010; accepted 05/05/2011.



Ramón Zatarain Cabada is a Professor and Researcher of the Master of Computer Science at the Instituto Tecnológico de Culiacán. He received a Master of Science and a PhD degrees in Computer Science from Florida Institute of Technology. He is a member of the National System of Researchers of

México, Level I. His main research interests include e-learning, m-learning and artificial intelligence in the education.



María Lucía Barrón Estrada has been a full-time Professor-Researcher in the Department of Computer Science at Instituto Tecnológico de Culiacán since 1992. In 2004 she received a Ph.D. in Computer Sciences from Florida Institute of Technology. She is recognized as a National

Researcher Level I in the National Research System in México. Dra. Barrón is the author and co-author of several publications in the area of learning using different approaches such as mobile learning, hybrid learning, etc., combined with artificial intelligence techniques in the development of authoring and learning tools to be used in education.



Carlos Alberto Reyes García is a PhD in Computer Sciences from the Florida State University. He is also MSc in Computer Sciences and MSc in Engineering Management from the Florida Institute of Technology. He is a full time researcher at the Department of Computer Science of

INAOE in Puebla, Mexico. He is a National Researcher Level I of the Mexican National System of Researchers.