EDUCA: A Web 2.0 Collaborative, Mobile and E-Learning Authoring System

Ramón Zatarain Cabada, María Lucía Barrón Estrada, Guillermo Sandoval Sánchez, Eduardo Urias Barrientos, Moisés Osorio Velázquez and Carlos Alberto Reyes García *
Instituto Tecnológico de Culiacán, Juan de Dios Batiz s/n Col. Guadalupe, C.P. 88220 Culiacán, México, tel. +52 667 7131796
* Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, C.P. 72840, México
kargaxxi@inaoep.mx
rzatarain@itculiacan.edu.mx

Abstract

In this paper, we present EDUCA, a Web 2.0 software tool to 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 artificial intelligence techniques like a neural network for selecting the best learning style or a recommendation-web mining system for adding and searching new learning resources.

1. Introduction

Many different author tools have been designed and implemented for fast and efficient production of web-based learning objects [1, 2]. However, all of those tools suffer from the same drawback or weakness. The author is the only one who defines and introduces the learning contents for any kind of course. With respect to this, in the last two or three years, a new generation of applications and tools named Web 2.0 applications have emerged. With this new technology, users not only retrieve information but also own and implement control over it.

In this context, we have designed and implemented a software tool (EDUCA) to create adaptive learning material in a Web 2.0 collaborative learning environment. The material is initially created by a tutor and later maintained and updated by the user/learner community to each individual course. The courses can dynamically recognize user learning characteristics and be displayed on mobile computers (cell phones, PDA, etc.). EDUCA makes use of Web 2.0 technologies like a recommendation system for filtering future Web learning resources and Web mining for discovering such learning resources [3].

2. Overall architecture

Figure 1 illustrates the overall architecture of EDUCA. As we can observe, there are two authors: the main tutor (a teacher or instructor) and the community of learners. When the authors add learning material, they first create four different instances corresponding to four different learning styles according to the Felder-Silverman model [4].

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 using a course. We also implemented a genetic algorithm (Bucket Sort) for the optimization of the weights used in the network. The network was trained for 800 generations using a population of 150 chromosomes. In order to train the network, we created three set of courses for high school students. Each course was presented in four different teaching styles according to the Felder-Silverman model. When a mobile 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.

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.
3. Intelligent Module

The Intelligent Module takes as an input the learning material for four different learning styles. Then, it creates a Neuro-Fuzzy Network (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 (it contains 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 representing the 7 linguistic variables used in the classification 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 follow:

\[ y_i^{(1)} = 0, \text{ if } x_i^{(1)} \leq a - b / 2 \]
\[ y_i^{(1)} = 1 - 2 |x_i^{(1)} - a| / b, \text{ if } a - b / 2 < x_i^{(1)} < a + b / 2 \]
\[ y_i^{(1)} = 0, \text{ if } x_i^{(1)} \geq a + b / 2 \]

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 a Centroid technique to make defuzzification. The value of the output is the learning style for the current student of the course.

Training the NFN with a Genetic Algorithm

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.

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.

4. Tests and Results

We tested the tool developing different kinds of courses like a GNU/Linux course, a Basic Math Operation course, and learning material for preparing to the Mexican University Admission test EXANI-II.

Next, we present an example of how an author creates/updates learning material for a Basic Math course (figure 2).

We first create the structure of the course (left-top). Then, we add learning material for each learning style (right-top and left-bottom). In this stage, we also assign fuzzy set values to each linguistic variable, and use
recommended and actual resources for inclusion in the course. Last, we export and display the course.

6. Related Work and Conclusions

There are tools used to create collaborative and web 2.0 learning systems like SHAREK [5], LearnHub [6], and WizIQ [7]. Those systems allow community learners to share resources for creating learning material. However the learning material is oriented to e-learning systems with no concerns to adaptation possibilities.

The work presented in this paper provides different aspects of our research. First, we illustrate an original architecture of the tool oriented to the Web 2.0 technology. Second, we discuss a novel technique for managing different styles of learning. Last, we show an exercise of crating learning materials using our tool.

![Figure 2. Creation of Learning Material for a Basic Math course](image)

**References**


