Oscar Castillo, Janusz Kacprzyk, and Witold Pedrycz (Eds.)

Soft Computing for Intelligent Control and Mobile Robotics

Studies in Computational Intelligence, Volume 318

Editor-in-Chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 294. Manuel F.M. Barros, Jorge M.C. Guilherme, and Nuno C.G. Horta
Analog Circuits and Systems Optimization based on Evolutionary Computation Techniques, 2010
ISBN 978-3-642-12345-0

Vol. 295. Roger Lee (Ed.)
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2010
ISBN 978-3-642-13264-3

Vol. 296. Roger Lee (Ed.)
Software Engineering Research, Management and Applications, 2010
ISBN 978-3-642-13272-8

Vol. 297. Tania Tronco (Ed.)
New Network Architectures, 2010
ISBN 978-3-642-13246-9

Vol. 298. Adam Wierzbicki
Trust and Fairness in Open, Distributed Systems, 2010
ISBN 978-3-642-13450-0

Vol. 299. Vassil Sgurev, Mincho Hadjiski, and Janusz Kacprzyk (Eds.)
Intelligent Systems: From Theory to Practice, 2010
ISBN 978-3-642-13427-2

Vol. 300. Baoding Liu (Ed.)
Uncertainty Theory, 2010
ISBN 978-3-642-13958-1

Vol. 301. Giuliano Armano, Marco de Gemmis, Giovanni Semeraro, and Eloisa Vargiu (Eds.)
Intelligent Information Access, 2010
ISBN 978-3-642-13999-4

Vol. 302. Bijaya Ketan Panigrahi, Ajith Abraham, and Swagatam Das (Eds.)
Computational Intelligence in Power Engineering, 2010
ISBN 978-3-642-14012-9

Vol. 303. Joachim Diederich, Cengiz Gunay, and James M. Hogan
Recruitment Learning, 2010
ISBN 978-3-642-14027-3

Vol. 304. Anthony Finn and Lakhmi C. Jain (Eds.)
Innovations in Defence Support Systems, 2010
ISBN 978-3-642-14083-9

Vol. 305. Stefania Montani and Lakhmi C. Jain (Eds.)
Successful Case-Based Reasoning Applications-1, 2010
ISBN 978-3-642-14077-8

Vol. 306. Tru Hoang Cao
Conceptual Graphs and Fuzzy Logic, 2010
ISBN 978-3-642-14086-0

Vol. 307. Anupam Shukla, Ritu Tiwari, and Rahul Kala
Towards Hybrid and Adaptive Computing, 2010
ISBN 978-3-642-14343-4

Vol. 308. Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi (Eds.)
Advances in Intelligent Tutoring Systems, 2010
ISBN 978-3-642-14362-5

Vol. 309. Isabelle Bichindaritz, Lakhmi C. Jain, Sachin Vaidya, and Ashlesha Jain (Eds.)
Computational Intelligence in Healthcare 4, 2010
ISBN 978-3-642-14463-9

Vol. 310. Dipti Srinivasan and Lakhmi C. Jain (Eds.)
Innovations in Multi-Agent Systems and Applications – 1, 2010
ISBN 978-3-642-14434-9

Vol. 311. Juan D. Velásquez and Lakhmi C. Jain (Eds.)
Advanced Techniques in Web Intelligence, 2010
ISBN 978-3-642-14460-8

Vol. 312. Patricia Melin, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
Soft Computing for Recognition based on Biometrics, 2010
ISBN 978-3-642-15110-1

Vol. 313. Imre J. Rudas, János Fodor, and Janusz Kacprzyk (Eds.)
Computational Intelligence in Engineering, 2010
ISBN 978-3-642-15219-1

Vol. 314. Lorenzo Magnani, Walter Carnielli, and Claudio Pizzi (Eds.)
Model-Based Reasoning in Science and Technology, 2010
ISBN 978-3-642-15222-1

Vol. 315. Mohammad Essaaidi, Michele Malgeri, and Costin Badica (Eds.)
Intelligent Distributed Computing IV, 2010
ISBN 978-3-642-15210-8

Vol. 316. Philipp Wolfrum
Information Routing, Correspondence Finding, and Object Recognition in the Brain, 2010
ISBN 978-3-642-15253-5

Vol. 317. xxx

Vol. 318. Oscar Castillo, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
Soft Computing for Intelligent Control and Mobile Robotics, 2010
ISBN 978-3-642-15533-8

Oscar Castillo, Janusz Kacprzyk, and
Witold Pedrycz (Eds.)

Soft Computing for Intelligent Control and Mobile Robotics

Springer

Prof. Oscar Castillo
Tijuana Institute of Technology
Department of Computer Science,
Tijuana, Mexico
Mailing Address
P.O. Box 4207
Chula Vista CA 91909, USA
E-mail: ocastillo@tectijuana.mx

Prof. Witold Pedrycz
University of Alberta,
Dept. Electrical and
Computer Engineering
Edmonton, Alberta T6J 2V4,
Canada
E-mail: pedrycz@ece.ualberta.ca

Prof. Janusz Kacprzyk
Polish Academy of Sciences,
Systems Research Institute,
Newelska 601-447 Warszawa Poland
E-mail: kacprzyk@ibspan.waw.pl

ISBN 978-3-642-15533-8 e-ISBN 978-3-642-15534-5

DOI 10.1007/978-3-642-15534-5

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2010934864

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

We describe in this book, hybrid intelligent systems using soft computing techniques for intelligent control and mobile robotics. Soft Computing (SC) consists of several intelligent computing paradigms, including fuzzy logic, neural networks, and bio-inspired optimization algorithms, which can be used to produce powerful hybrid intelligent systems. The book is organized in five main parts, which contain a group of papers around a similar subject. The first part consists of papers with the main theme of theory and algorithms, which are basically papers that propose new models and concepts, which can be the basis for achieving intelligent control and mobile robotics. The second part contains papers with the main theme of intelligent control, which are basically papers using bio-inspired techniques, like evolutionary algorithms and neural networks, for achieving intelligent control of non-linear plants. The third part contains papers with the theme of optimization of fuzzy controllers, which basically consider the application of bio-inspired optimization methods to automate the design process of optimal type-1 and type-2 fuzzy controllers. The fourth part contains papers that deal with the application of SC techniques in times series prediction and intelligent agents. The fifth part contains papers with the theme of computer vision and robotics, which are papers considering soft computing methods for applications related to vision and robotics.

In the part of theory and algorithms there are 5 papers that describe different contributions that propose new models and concepts, which can be the considered as the basis for achieving intelligent control and mobile robotics. The first paper, by Ramon Zatarain et al., deals with applying intelligent systems for modeling students' learning styles used for mobile and web-based systems. The second paper, by Luis Martinez et al., deals with a fuzzy model for RAMSET: Role Assignment Methodology for Software Engineering Teams. The third paper, by Jorge Soria-Alcaraz et al., describes an academic timetabling design using hyper-heuristics. The fourth paper, by Alberto Ochoa et al., describes a logistics optimization service improved with artificial intelligence. The fifth paper, by Francisco Arce and Mario Garcia-Valdez, describes an accelerometer-based hand gesture recognition system using artificial neural networks.

In the part of intelligent control there are 5 papers that describe different contributions on achieving control using hybrid intelligent systems based on soft computing techniques. The first paper, by Ieroham Baruch et al., describes a direct and indirect neural identification and control of a continuous bioprocess via Marquardt learning. The second paper, by Eduardo Gomez-Ramirez et al., deals with a method for simple tuning of type-2 fuzzy controllers. The third paper, by Leocundo Aguilar et al., proposes an increasing energy efficiency of a preamble sampling MAC protocol for wireless sensor networks using a fuzzy logic approach. The fourth paper, by Arnulfo Alanis et al., describes a multi-agent system based on psychological models for mobile robots. The fifth paper, by Fevrier Valdez et al., proposes the use of fuzzy logic to control parameters in bio-inspired optimization methods.

In the part of optimization of fuzzy controllers there are 5 papers that describe different contributions of new algorithms for optimization and their application to designing optimal fuzzy logic controllers. The first paper by Ricardo Martinez et al., describes the optimization of type-2 fuzzy logic controllers using PSO applied to linear plants. The second paper, by Yazmin Maldonado et al., deals with an approach for the optimization of membership functions for an incremental fuzzy PD control based on genetic algorithms. The third paper, by Leticia Cervantes and Oscar Castillo, describes a new method for the design of a fuzzy system for the longitudinal control of an F-14 airplane. The fourth paper by Abraham Melendez et al., describes a fuzzy reactive controller of a mobile robot. The fifth paper, by Arnulfo Alanis et al., describes a multi-agent system with personality profiles and preferences and learning for autonomous mobile robot, with fuzzy logic support.

In the part of time series prediction and intelligent agents several contributions are described on the development of new models and algorithms relevant to time series analysis and forecasting, as well as the application of intelligent agents in real-world applications. The first paper, by Pilar Gomez et al., describes composite recurrent neural networks for long-term prediction of highly-dynamic time series supported by wavelet decomposition. The second paper, by Juan R. Castro et al., describes an interval type-2 fuzzy neural network for chaotic time series prediction with cross-validation and the Akaike test. The third paper, by Jesus Soto et al., deals with chaotic time series prediction using Ensembles of ANFIS. The fourth paper, by Lucila Morales et al., describes the modeling of facial expression of intelligent virtual agents. The fifth paper, by Ivan Espinoza et al., describes agent communication using semantic networks. The sixth paper, by Cecilia Leal-Ramirez et al., describes a fuzzy cellular model for predator-prey interaction applied to the control of plagues in a peppers cropping.

In the part of computer vision and robotics several contributions on models and algorithms are presented, as well as their applications to different real-world problems. The first paper, by Rogelio Salinas-Gutierrez et al., describes the use of Gaussian copulas in supervised probabilistic classification. The second paper, by Pablo Rivas et al., proposes subjective co-localization analysis with fuzzy predicates. The third paper, by Jesus David Teran et al., describes an iterated local search algorithm for the linear ordering problem with cumulative costs. The fourth paper, by Nohe Cazarez et al., describes an observer for the type-1 fuzzy control of a servomechanism with backlash using only motor measurements. The fifth paper, by Selene Cardenas et al., proposes a neuro-fuzzy based output feedback controller design for biped robot walking. The sixth paper, by Oscar Montiel et al., describes a fuzzy system to control the movement of a wheeled mobile robot. The seventh paper, by Oscar Montiel et al., proposes an approach for embedding a fuzzy locomotion pose controller for a wheeled mobile robot into an FPGA.

In conclusion, the edited book comprises papers on diverse aspects of bio-inspired models, soft computing and hybrid intelligent systems for control and mobile robotics. There are theoretical aspects as well as application papers.

May 31, 2010

Oscar Castillo
Tijuana Institute of Technology, Mexico

Witold Pedrycz
University of Alberta, Canada

Janusz Kacprzyk
Polish Academy of Sciences, Poland

Contents

Part I: Theory and Algorithms

Applying Intelligent Systems for Modeling Students' Learning Styles Used for Mobile and Web-Based Systems 3
Ramón Zatarain, Lucia Barrón-Estrada, Carlos Alberto Reyes-García, Orion Fausto Reyes-Galaviz

Towards a Fuzzy Model for RAMSET: Role Assignment Methodology for Software Engineering Teams 23
Luis G. Martínez, Juan R. Castro, Guillermo Licea, Antonio Rodríguez-Díaz, Carlos Álvarez

Academic Timetabling Design Using Hyper-Heuristics 43
Soria-Alcaraz Jorge A., Carpio-Valadez J. Martin, Terashima-Marin Hugo

Logistics Optimization Service Improved with Artificial Intelligence 57
Alberto Ochoa, Yazmani Garcia, Javier Yañez

Accelerometer-Based Hand Gesture Recognition Using Artificial Neural Networks 67
Francisco Arce, José Mario García Valdez

Part II: Intelligent Control

Direct and Indirect Neural Identification and Control of a Continuous Bioprocess via Marquardt Learning 81
Ieroham Baruch, Carlos-Roman Mariaca-Gaspar, Josefina Barrera-Cortes, Oscar Castillo

Simple Tuning of Type-2 Fuzzy Controllers 103
Eduardo Gómez-Ramírez, Patricia Melin, Oscar Castillo

Increasing Energy Efficiency of a Preamble Sampling MAC Protocol for Wireless Sensor Networks Using a Fuzzy Logic Approach 125
Leocundo Aguilar, Oscar Castillo, J. Antonio García-Macías, Guillermo Licea

Multi-Agent System Based on Psychological Models for Mobile Robots 143
Arnulfo Alanis Garza, Oscar Castillo, José Mario García Valdez

Fuzzy Control for Dynamical Parameter Adaptation in a Parallel Evolutionary Method Combining Particle Swarm Optimization and Genetic Algorithms 161
Fevrier Valdez, Patricia Melin, Oscar Castillo

Part III: Optimization of Fuzzy Controllers

Optimization of Type-2 Fuzzy Logic Controllers Using PSO Applied to Linear Plants 181
Ricardo Martinez, Oscar Castillo, Luis T. Aguilar, Antonio Rodriguez

Optimization of Membership Functions for an Incremental Fuzzy PD Control Based on Genetic Algorithms 195
Yazmín Maldonado, Oscar Castillo, Patricia Melin

Design of a Fuzzy System for the Longitudinal Control of an F-14 Airplane 213
Leticia Cervantes, Oscar Castillo

A Fuzzy Reactive Controller of a Mobile Robot 225
Abraham Meléndez, Oscar Castillo, Arnulfo Alanis Garza

Multi-Agent System with Personality Profiles and Preferences and Learning for Autonomous Mobile Robot with Fuzzy Logic Support 233
Arnulfo Alanis Garza, Oscar Castillo, José Mario García Valdez

Part IV: Time Series Prediction and Intelligent Agents

Composite Recurrent Neural Networks for Long-Term Prediction of Highly-Dynamic Time Series Supported by Wavelet Decomposition 253
Pilar Gomez-Gil, Angel Garcia-Pedrero, Juan Manuel Ramirez-Cortes

An Interval Type-2 Fuzzy Neural Network for Chaotic Time Series Prediction with Cross-Validation and Akaike Test 269
Juan R. Castro, Oscar Castillo, Patricia Melin, Olivia Mendoza, Antonio Rodríguez-Díaz

Chaotic Time Series Prediction Using Ensembles of ANFIS 287
Jesus Soto, Oscar Castillo, José Soria

Modeling Facial Expression of Intelligent Virtual Agents 303
María Lucila Morales-Rodríguez, Fabián Medellín-Martínez, Juan J. González B.

Agent Communication Using Semantic Networks 315
Iván Espinoza-Hernández, Dora-Luz Flores, Antonio Rodríguez-Díaz, Manuel Castañón-Puga, Carelia Gaxiola

Fuzzy Cellular Model for Predator-Prey Interaction Applied to the Control of Plagues in a Peppers Cropping 329
Cecilia Leal-Ramirez, Oscar Castillo, Antonio Rodriguez-Diaz

Part V: Vision and Robotics

Using Gaussian Copulas in Supervised Probabilistic Classification 355
Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, Mariano J.J. Rivera-Meraz, Enrique R. Villa-Diharce

Subjective Colocalization Analysis with Fuzzy Predicates 373
Pablo Rivas-Perea, Jose Gerardo Rosiles, Wei Qian

Iterated Local Search Algorithm for the Linear Ordering Problem with Cumulative Costs (LOPCC) 395
Jesús David Terán Villanueva, Héctor Joaquín Fraire Huacuja, Rodolfo Pazos Rangel, Juan Martín Carpio Valadez, Héctor José Puga Soberanes, Juan Javier González Barbosa

An Observer for the Type-1 Fuzzy Control of a Servomechanism with Backlash Using Only Motor Measurements 405
Nohe R. Cazarez-Castro, Luis T. Aguilar, Oscar Castillo

Neuro-Fuzzy Based Output Feedback Controller Design for Biped Robot Walking 423
Selene L. Cardenas-Maciel, Oscar Castillo, Luis T. Aguilar

Fuzzy System to Control the Movement of a Wheeled Mobile Robot 445
Oscar Montiel Ross, Jesús Camacho, Roberto Sepúlveda, Oscar Castillo

Embedding a Fuzzy Locomotion Pose Controller for a Wheeled Mobile Robot into an FPGA 465
Oscar Montiel, Jesús Camacho, Roberto Sepúlveda, Oscar Castillo

Author Index 483

Part I
Theory and Algorithms

Applying Intelligent Systems for Modeling Students' Learning Styles Used for Mobile and Web-Based Systems

Ramón Zatarain[1], Lucia Barrón-Estrada[1], Carlos Alberto Reyes-García[2], and Orion Fausto Reyes-Galaviz[3]

[1] Instituto Tecnológico de Culiacán, México
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), México,
[3] Universidad Autónoma de Tlaxcala
{rzatarain,lbarron,vponce,rcabada}@itculiacan.edu.mx,
kargaxxi@inaoep.mx, orionfrg@yahoo.com

Abstract. The identification of the best learning style in an Intelligent Tutoring System must be considered essential as part of the success in the teaching process. This research work presents a set of three different approaches applying intelligent systems for automatic identification of learning styles in order to provide an adapted learning scheme under different software platforms. The first approach uses a neuro-fuzzy network (NFN) to select the best learning style. The second approach combines a NFN to classify learning styles with a genetic algorithm for weight optimization. The learning styles are based on Gardner's Pedagogical Model of Multiple Intelligences. The last approach implements a self-organising feature map (SOM) for identifying learning styles under the Felder-Silverman Model. The three approaches are used by an author tool for building Intelligent Tutoring Systems running under a Web 2.0 collaborative learning platform. The tutoring systems together with the neural networks can also be exported to mobile devices. We present results of three different tutoring systems produced by three implemented authoring tools.

Keywords: Intelligent Tutoring System, Web 2.0, Authoring Tool, M-Learning.

1 Introduction

In an e-learning environment, teachers can design various alternatives of content for each possible configuration of learning style [1]. The characteristics of e-learning allow teachers to work collaboratively with students, and even with other teachers interested in designing learning materials. Thus, responsibility for the design of teaching material does not lie with one person, but on a community of designers [2]. The time devoted to teaching every student is not a problem in electronic learning environments. Each student can have access to materials virtually

O. Castillo et al. (Eds.): Soft Computing for Intell. Control and Mob. Robot., SCI 318, pp. 3–22.
springerlink.com

at the time and place he wants [3]. An implementation of e-learning, in which materials are selected according to student's learning style, can be defined in the context of an Intelligent Tutoring System (ITS) or in an Adaptive Hypermedia System. An ITS refers to any computer system that provides a personalized educational process directly to the student. The ITS works automatically and requires no intervention by the teacher (or student) to perform the customization of the learning material. To identify the learning style of students there are several solutions. One solution is the application of questionnaires designed by teachers specialized in the field. Another more accurate solution is to use artificial intelligence techniques for carrying out this process automatically.

This work addresses the above problems through design and construction of a neural network, which after training, automatically and dynamically identify student learning styles. This neural network provides the means for customizing learning materials. The neural network can be used in any e-learning or m-learning environment, whether for identification of learning style in the context of an Intelligent Tutoring System, or manually with the student consulting its own learning styles.

The arrangement of the paper is as follows: Section 2 gives a general structure of the tool. Section 3 presents the neural network and predictive engine used in the tool. Results are shown in Section 4. Discussions and conclusions are given in Section 5.

2 Learning Style Models

Models of learning styles categorized both the ways in which students learn and how teachers teach. Its main objective is that in each category holding the model, the learning needs of students are met [4]. Taking into account the majority of existing models concerning learning styles, there are five major families [5]. Figure 1 shows the first four classes of models, which relate to theories of learning styles, the fifth family, which is omitted in the figure, contains models that deviate from the concept of learning styles and propose other theories.

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, and provides that each individual has different intelligences at different levels. Gardner also states that every person has a unique cognitive profile. According to multiple intelligence theory, there are nine basic types of intelligence: 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 conforms to all subjects. The scales are extraversion / introversion, sensitive / intuitive, thinking / feeling, and judgment / perception. Of 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 [6]. 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**
Bartlett	Broverman	Apter	Herrmann
Betts	Cooper	Epstein y	Kolb
Dunn Dunn	Guilford	Meier	Allison y
Gordon	Hulzman y	Harrison-	Hayes
Marks	Hudson	Branson	Honey y
Paivio	Hunt	Jackson	Mumford
Richardson	Kagan	Miller	Kaufmann
Sheehan	Kogan		Kirton
Torrance	Messick		McCarthy
	Pettigrew		

Fig. 1. Main Learning Style Models

3 First Approach: Modeling Learning Styles with a NFN

The first approach to identify students' learning styles was using a neuro-fuzzy network. In this section, we present **MLTutor**, an authoring tool which can be used to build personalized or intelligent tutoring courses to be used in both learning settings: distance and mobile learning. The tool can be employed for developing learning material using SCORM [7] Learning Objects or other standard file formats. The output of the authoring tool will be either learning material for mobile devices or SCORM learning objects for e-learning environments. 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 [8].

3.1 MLTutor Architecture

Figure 2 presents the architecture of **MLTutor**. As shown in Figure 2, the tool has two main editors: the **content editor** and the **fuzzy set editor**. An author creates a tutoring system by first building a course structure using the content editor. This structure consists of a number of units where a unit can be linked with learning material and some assessments. A course is created by importing already prepared learning material in different standard formats like **html, pdf, doc** or **SCORM** learning objects from any type of source. The author can also introduce learning material by using a simple editor included in the tool. Other important learning materials the authors insert into each one of the units are quizzes. These can be in every part of each section. The quiz is essential for the dynamic courseware generation because from the test results, the neural networks classify learning styles.

When the author introduces the learning material he/she creates four different instances corresponding to four different student learning styles (types) 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 helps the user to define **Fuzzy Membership Functions**. For **Fuzzy Inputs** there are seven linguistic variables defined for a user. They are: *answer selection order, correct answers, quiz spent time, topic spent time, number of tries until correct answer, number of visits to a question, and number of visits to a topic*. Each of the seven linguistic variables allows three different values: low, average, and high. A Fuzzy inference process is performed mapping 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** [9]. It operates online using present and former information for each individual learner. At the start of each learning unit, predictions are made as to what the learners preferred learning material is, based on former information built in the fuzzy set edition phase. Afterward, 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 job of displaying the material of the course into the mobile device, according to some chosen learning style.

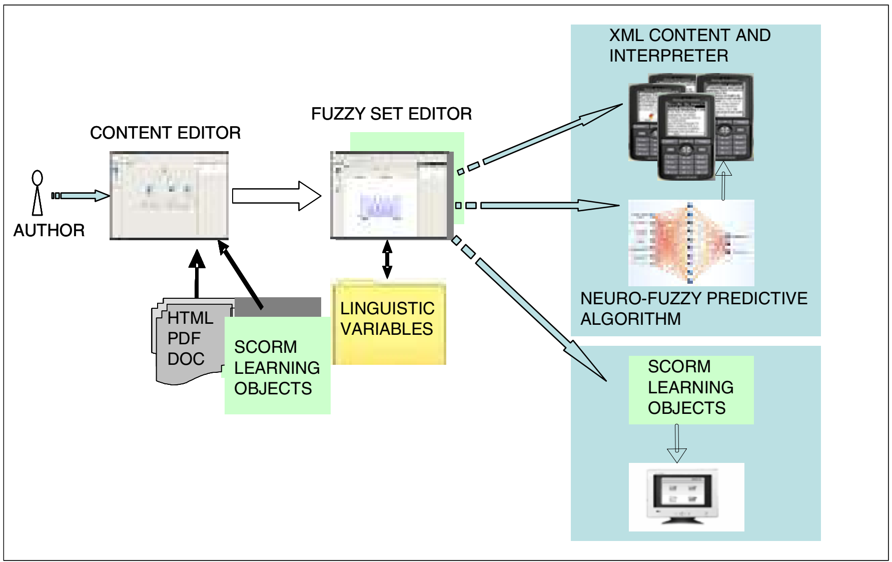


Fig. 2. MLTutor Architecture

3.2 Neuro-fuzzy Predictive Model

Figure 3 shows part of the MLTutor Neuro-Fuzzy system (just two linguistic variables are shown) implemented in order to represent knowledge (seven linguistic variables, four multiple intelligences, and inference rules), to learn from former and current data, and to make adjustment to new learning styles.

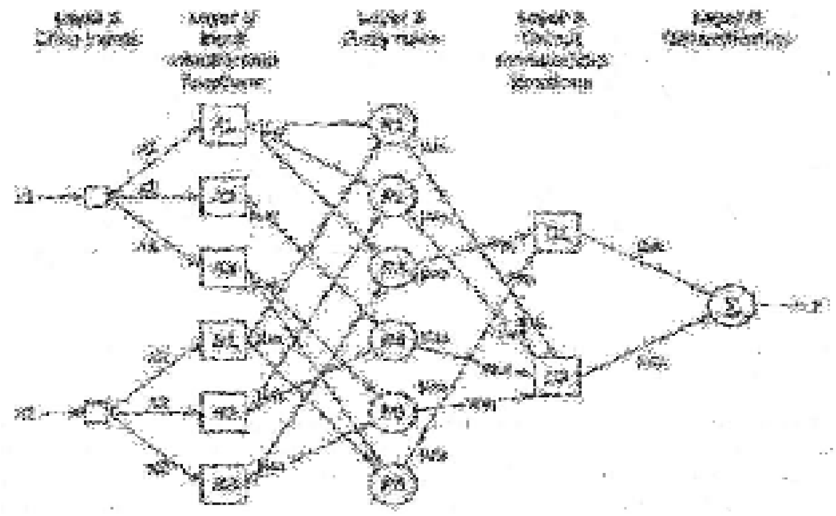


Fig. 3. Part of the HLTutor Neuro-Fuzzy System

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. Every linguistic variable has three fuzzy sets (low, average, and high). The input layer sends out external crisp values directly to the next layer. The output of this layer is as follows:

$y_i = x_i$,

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 crisp values from input layer are transformed to appropriate 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 the layer is the degrees of membership 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:

$y_i = x_{1i} \times x_{2i} \times \ldots \times x_{ki}$

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 on many adjustments, by dividing each weight into the greatest weight found on each one of the adjustments or iterations and is represented by

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

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.

Table 1. Fuzzy sets for two Linguistic Variables

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

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

$x_{1i} \oplus x_{2i} \oplus \ldots \oplus x_{ji}$

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 **defuzzyfication layer** is the output of our system. In this layer the **sum-product composition** method is used [10]. The output of this last layer is a recommended learning style for the student. The learning algorithm we used is **back-propagation**. Our algorithm takes a desired output (a default learning style), which is computed at the beginning of the iterations. Next, the algorithm computes the actual output and compares it with the desired output (which is 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 backward through the network, updating or adjusting the weights between connected neurons. The neural network was trained too using Matlab (version 7.1). Then, the trained neural network was implemented using Java along with the XML interpreter.

3.3 Tests and Results

We conducted a test of our tool by working with a group of different kinds of users (authors), mainly university professors, and college students. They produced personalized or intelligent tutoring courses for English Language (Elementary school), Object-Oriented Analysis and Design, and a Basic Math Course. The courses dynamically adapt the content or style of instruction presented to the learner. Figure 4 shows some interfaces (snapshots) of the authoring tool when building the Basic Math course. As we can observe, one image shows the course tree structure (left-top); two images show the contents of the course under the Verbal/Linguistic type of Intelligence (right-top and left-bottom); another one illustrates the fuzzy set edition (right-bottom); and the last one (front image) illustrates a quiz edition.

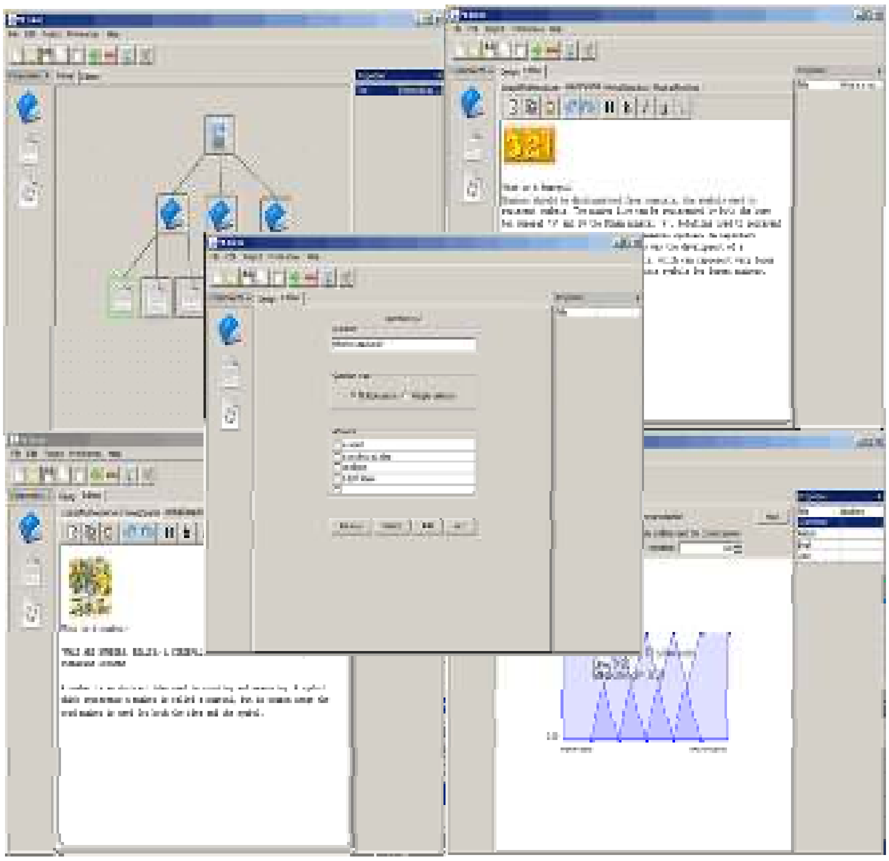


Fig. 4. Interfaces of MLTutor and a Math Course

Figure 5 presents some instances of the Math course running in a cell Phone (NHAL Win32 Emulator). The four mobile instances correspond (left-to-right, top-bottom) to *Three-Chapter menu, Chapter-1 menu (contents and quiz), chapter-1 contents, and chapter-1 quiz.*

Fig. 5. The Math Course in a Cell Phone

4 Second Approach: A NFN and a Genetic Algorithm

The second approach to identify students' learning styles was using a neuro-fuzzy network together with a Genetic Algorithm. In this section, 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 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 6 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**. The student or learner is an important author of the course and participate actively adding learning resources to the courses. The learner has a user profile with information like the GPA, 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 [6].

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** 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.

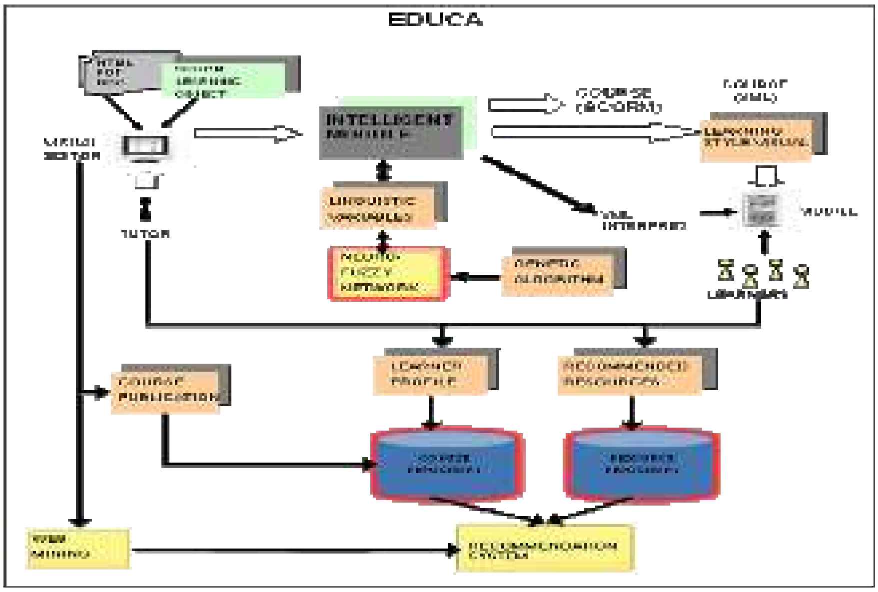


Fig. 6. EDUCA Architecture

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 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 (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.

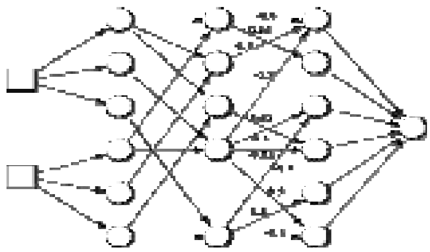
The first layer of the NFN has 7 neurons (see figure 3) representing the 7 linguistic variables used in the classification of the learning style. Every neuron of layer 1 is connected to 3 neurons of layer 2 (*fuzzyfication layer*). Due to the fact that we are using triangular membership functions, the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

$y_i^{(1)} = 0$, if $x_i^{(1)} \leq a - b / 2$
$y_i^{(1)} = 1 - 2\,|x_i^{(1)} - a| / b$, if $a - b / 2 < x_i^{(1)} < a + b / 2$
$y_i^{(1)} = 0$, 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 *defuzzyfication*. The value of the output is the learning style for the current student of the course.

Training the NFN with a Genetic Algorithm



At the beginning, we create a chromosome population with random values. Each decoded element of the population, represent the weights in the NFN (see figure 7).

-0.6	0.5	-0.3	0.14	0.3	-0.4	0.63	-0.21	0.8	0.9	-0.1

Fig. 7. Decoding the weights in the NFN into a chromosome

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.

In order to train the network, we create three set of courses for high school students. Each course was presented in eight different styles for 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 *Felder-Silverman Index of Learning Style Questionnaire* to every student. We also designed an exam 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 taken course (randomly chosen). 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 run any course stored in the mobile by reading the XML file where the course is stored. At the beginning, a learning style is assigned to the student or user of the course. Then, depending of the results from questions prepared to the students inside the learning objects, the learning material is adapted to the best learning style (Visual, Verbal, Sequential, etc.) of the student (the NFN is consulted in order to find the best appropriated learning style to the student).

Fig. 8. Dynamic sequencing of Learning Material

An important feature is the possibility of tracing the different learning styles that a user or student follow during lesson learning. The interpreter of the course optionally displays the current student learning styles 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 8 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).

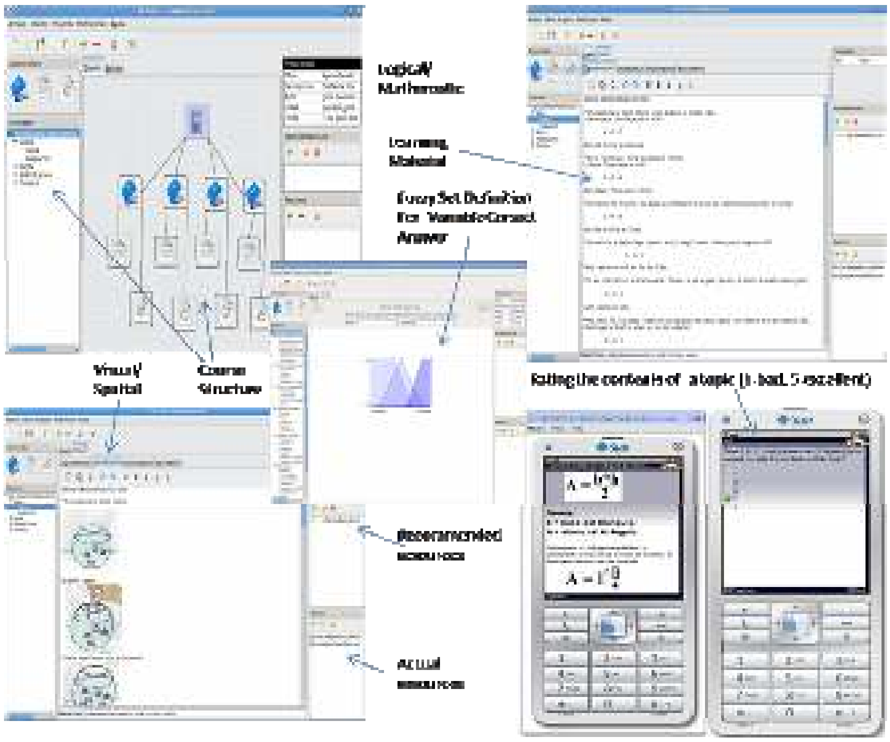


Fig. 9. Creation of Learning Material for a Basic Math course

4.4 Test and Results

We tested the tool with 15 professors/teachers and their respective students of different teaching levels. They developed different kinds of courses like a GNU/Linux course, a Basic Math Operation course, and learning material for preparation to the University Admission test EXANI-II. Each one of the courses had from 4 to 10 units of learning. They included evaluations on each unit. The students participated by reading, evaluating and adding material (Web resources) to the courses. We applied a questionnaire or survey to instructors and students in order to evaluate the effectiveness of **EDUCA**. More than 90% of them agree that they would like to use the tool for teaching their courses. This point is very important because the tool was developed to be used by instructors of any level of teaching (from elementary to college level).