World Journal on Educational Technology

2 (2009) 78-88

www.world-education-center.org/index.php/wjet

Modeling student's learning styles in web 2.0 learning systems

Ramón Zatarain-Cabada [a] *, M. L. Barrón-Estrada [a], L. Zepeda-Sánchez [a], Guillermo Sandoval [a],
J.M. Osorio-Velazquez [a], J.E. Urias-Barrientos [a]

[a] Instituto Tecnológico de Culiacán
Juan de Dios Bátiz s/n, col. Guadalupe, Culiacán, Sinaloa, 80220, México

Received July 21, 2009; revised September 08, 2009; accepted October 22 , 2009

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. In many implementations of automatic classifiers finding the right student learning style represents the hardest assignment. The reason is that most of the techniques work using expert groups or a set of questionnaires which define how the learning styles are assigned to students. This paper presents a novel approach for automatic learning styles classification using a Kohonen network. The approach is 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 network can also be exported to mobile devices. We present different results to the approach working under the author tool.

Keywords: intelligent tutoring system; web 2.0; authoring tool; w-learning

1. INTRODUCTION

Learner or Student Models are the core of the personalization of Intelligent Tutoring Systems. They make available tutoring tailored or adapted to the needs of individual students (Kerly, 2008). Many approaches and implementations have been developed in recent years in order to model

* Ramón Zatarain-Cabada. Tel.: +52 667 7131796
E-mail address: rzatarain@itculiacan.edu.mx

students' learning styles (Carmona *et al*, 2008; Graf *et al*, 2008). Most of those implementations use Bayesian Networks (Carmona *et al*, 2008), Linear Temporal Logic (Limongelli *et al*, 2008), or neuro-fuzzy networks (Zatarain-Cabada *et al*, 2008). In the case of using a learning model like Felder-Silverman (Felder & Silverman, 1988), we also use the Index of Learning Style Questionnaire (ILSQ) (Felder & Solomon, 2004).

In this work we propose a different approach for selecting a student´s learning style using self-organising feature maps (Kohonen neural networks) (Kohonen, 2001). Advantages of Kohonen networks include implementation simplicity, execution speed and a shorter training process; however maybe the most important advantage of these unsupervised neural networks is that they do not require an external teacher for presenting a training set. During a training session, our Kohonen network receives a number of different input patterns (the student learning style obtained from the ILSQ, 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 input.

Recently many research groups in the field of education are using Web 2.0 technologies, such as wikis, blogs, recommendation systems and social networking (O´Reilly, 2005). The e-learning tutor centered is shifting to become more learner centered, where learners are also part of a community of authors and users of the learning resources (Hage and Aimeur, 2008). 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/instructor 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 as a recommendation system for filtering future Web learning resources, and Web mining for discovering such resources.

The arrangement of the paper is as follows: Section 2 describes the general structure of the tool. Section 3 presents the neural network used in the tool. Tests and results are shown in Section 4. Comparison to related work is given in section 5. Conclusions and future work are discussed in Section 6.

2. METHOD

The methodology for the process of identifying learning styles was divided into two stages: a stage of training the neural network and a stage of application. In the first stage, the neural network is prepared to be able to identify learning styles of students. In the second stage, the neural network is used as part of the module for mobile devices of EDUCA.

The data space used for training the network was collected through a process involving fieldwork. This study included 47 tests with students of upper secondary education. Each student was provided with the Questionnaire Learning Styles Inventory of Felder-Soloman (Felder & Solomon, 2004). The results of the survey identified and recorded the learning style of each student. Educational materials were developed for three different topics: photography, eolic energy and computing introduction.

From each of these topics we created 8 different versions. Each of the versions of each course was modeled using a different learning style.

Since we only took into consideration 3-dimensional learning styles of Felder-Silverman, we implemented 8 different learning styles. The eight different learning styles are the result of the combination of visual-verbal, sensitive- intuitive and sequential-global dimensions. Each student was provided with a version of each course. The version of each course was determined randomly. The student was allowed to study the material for a period of approximately 40 minutes.

For the stage of application of the neural network, we proceeded to include this within a program (an interpreter) which displays the courses in mobile devices; such courses were first created with the authoring tool EDUCA. By using the neural network, the interpreter identifies the best learning style of the user.

2.1 The Authoring Tool

Figure 1 presents the general structure of the authoring tool Educa. As shown in Figure 1, there are two main authors: the instructor and the learners. An instructor creates an intelligent tutoring (adaptive) system by first building a knowledge base (a learning object container) using a visual editor. 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 insert learning material by using a simple editor included in the tool. The learning material is classified into four different types of learning objects: text, image, audio, and video.

In each section the author inserts learning objects and defines a Felder-Silverman Result Score (FSRS) for every dimension: Input (visual/verbal), Perception (sensitive/intuitive), Understanding (sequential/global) and Processing (active/reflective). The scores defined by the authors will decide what learning objects are presented to the students depending of their learning style. Another way to understand this approach is that the authors create a Knowledge set which contains all the possible learning objects in different learning styles; the learners only "see" part of the set (a knowledge subset) depending of their own learning styles. Another important learning object the authors insert into the knowledge repository is a quiz. 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.

After the author create the knowledge base of the ITS, 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. Another option to the output of the visual 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.

The other author in Educa is the Learners. When a student or learner reads a course or ITS for the first time, a user profile is created in the system. This profile contains student data like student's learning style, academic records and GPA, past uploaded resources (recommendation), and general academic information.

Once an ITS 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 ITS 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 (Burke, 2002) which stores new resources into a Resource Repository. Lastly, Educa uses a Data or Text Mining Subsystem used to search resources in the Web. All of the modules or subsystems in the bottom middle part of Educa (User Profile, Recommendation System, Data mining, and Repositories) stand for the Web 2.0 Collaborative Learning System of Educa.

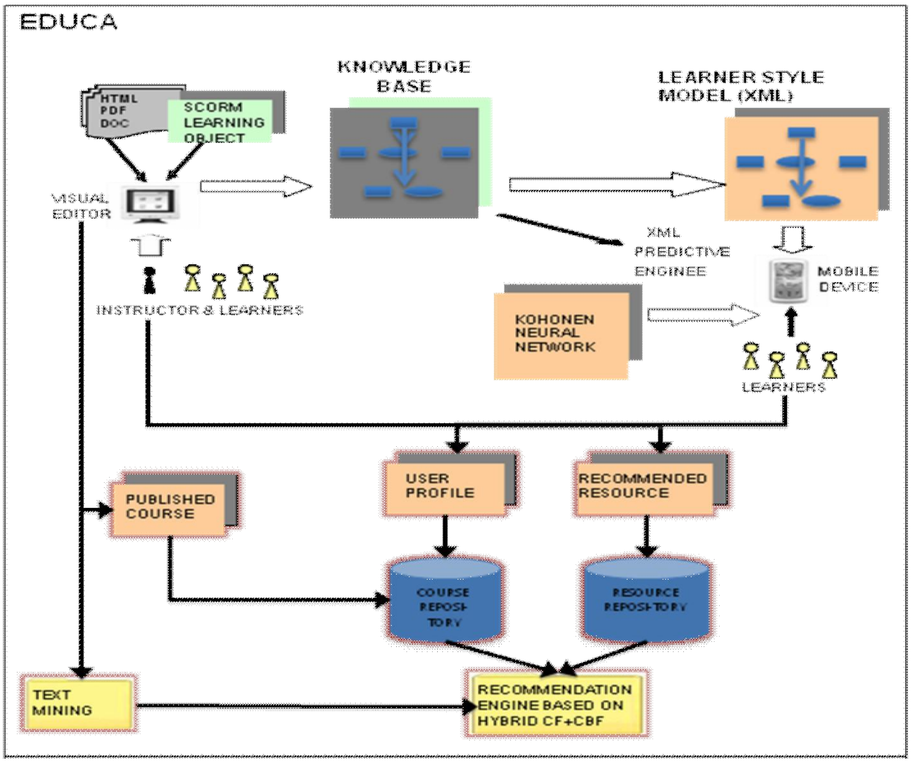


Figure 1. *General structure of Educa*

2.2 Educa General Structure

A self-organizing feature map or Kohonen neural network is trained using unsupervised or competitive learning (Kohonen, 2001). In competitive learning, neurons compete among themselves to be activated. The Kohonen model provides topological mapping, placing a fixed number of input patterns from the input layer into a higher-dimension output or Kohonen layer (figure 2). The layer consists of a single layer of computation neurons, with two types of different connections: forward connections, from input layer to the layer, and lateral connections between neurons in the layer. During training, the neural network receives new input patterns presented as vectors. Every neuron in the Kohonen layer receives a modified copy of the input pattern. The lateral connections produce excitatory or inhibitory effects, depending on the distance from the winning neuron.

2.3 The Neural Network

For the development of this research we chose a self-organizing map (known also as Kohonen networks). The choice of this type of neural networks is motivated mainly by three factors. The first factor is the type of learning (unsupervised) of such networks. The second is network performance and the third one is training speed.

Identifying learning styles of students is a complex task of pedagogy. The proper training of a neural network with supervised learning requires, in most cases, a pedagogy expert. By using self-organizing maps we do not need the specialist in the training process.

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.

An input vector in our neural network is described as:

$$D = [d_{fs}\ d_a\ p] \qquad (1)$$

where D is the input vector, and it is formed by three elements: d_{fs} which is the student learning style identified by applying the Felder-Silverman ILSQ questionnaire; d_a which 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 has learning style *dfs* and the course was offered using learning style $d_{a.}$

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

$$d_a = d_{fs} = [c_1\ c_2\ c_3] \quad (2)$$

where $c_1\ c_2\ c_3$ represents three scores for Perception, Input, and Understanding Dimension in the ILSQ questionnaire.

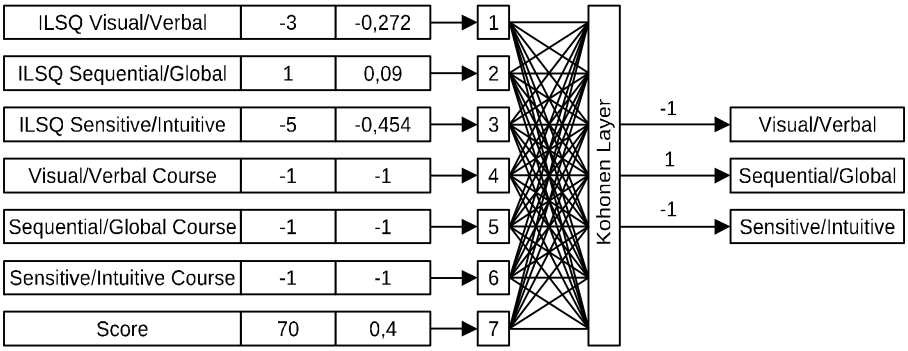
Figure 2 shows part of the Kohonen network architecture. We can see that the input layer has seven neurons, corresponding to three-dimension vectors d_{fs} and d_a, and the student grade p. Vector's input data vary between -11 to +11, which is massaged to the range -1 to +1 with Eq. 1.

$$\text{Massaged value}_{11} = v_i \ / \ |v_{imax}| \qquad (3)$$

where v_i is a ILSQ score, and v_{imax} is the maximum ILSQ score value (+11). In figure 2, as we observe, -3, an ILSQ score for visual/verbal learning style, is mapped to -0.272. On the other hand, the student grade, which varies between 0 and 100, is also massaged to the range -1 to +1, with Eq. 2.

$$v_{nj} = v_j \times 2 / v_{jmax} - 1 \qquad (4)$$

where v_{nj} is the new ILSQ score, v_j is the current ILSQ score, and v_{jmax} is the maximum student grade (+100). For example, 70 is mapped to 0.4 (figure 2).



ILSQ Visual/Verbal	-3	-0,272
ILSQ Sequential/Global	1	0,09
ILSQ Sensitive/Intuitive	-5	-0,454
Visual/Verbal Course	-1	-1
Sequential/Global Course	-1	-1
Sensitive/Intuitive Course	-1	-1
Score	70	0,4

Figure 2. *The Kohonen network for learning styles*

2.4 Using the Kohonen Network

As we explained before the network is applied to identify learning styles of students. The network is then exported together with the ITS and the predictive engine to the mobile (see figure 1). When the student is displaying an ITS in a time *t*, the learning material shown to the students is using a learning style *g*. Whenever the student answers a quiz as part of the learning contents, 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 (Eq. 5):

$$j_x (p) = \min ||X - W_j (p)|| = [\Sigma(x_i - w_{ij})^2]^{1/2} \qquad (5)$$

where *n* is the number of neurons in the input layer, *m* is the number of neurons in the Kohonen layer, *X* is the input pattern vector and *W* is the weight vector. After, finding neuron j_x, the output of the network is the student learning style which is defined with Eq. 6.

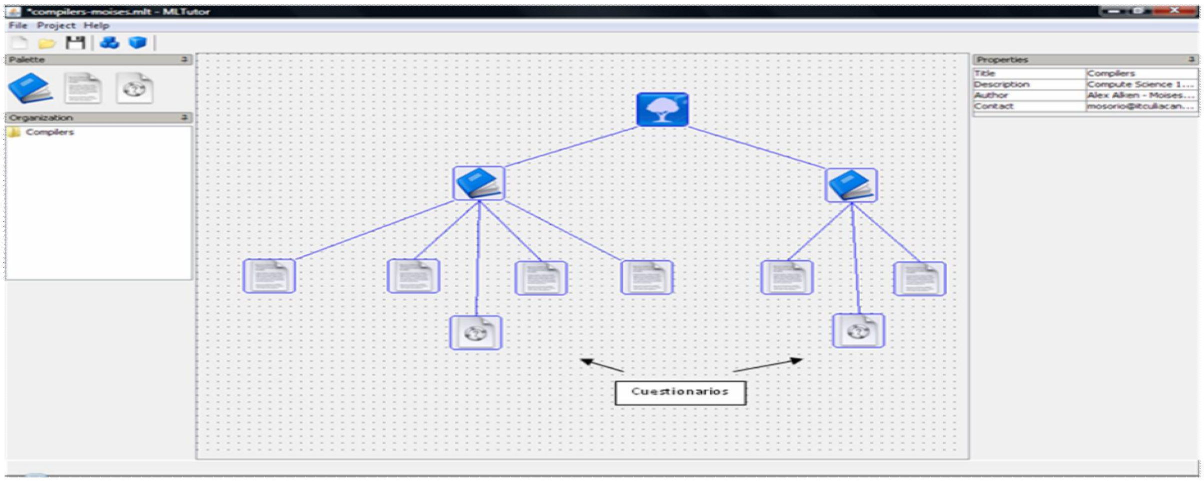
$$g(t + 1) = r(g(t), k) \qquad (6)$$

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

3. RESULTS AND DISCUSSIONS

The evaluation of our software is given from two different perspectives: first an assessment of the authoring tool, in this case EDUCA, which is evaluated with parameters such as software interface usability, time to learn the tool, speed of performance and so on. And second, an evaluation of the tutors generated by the tool, which in our case are adaptive or intelligent tutoring systems.

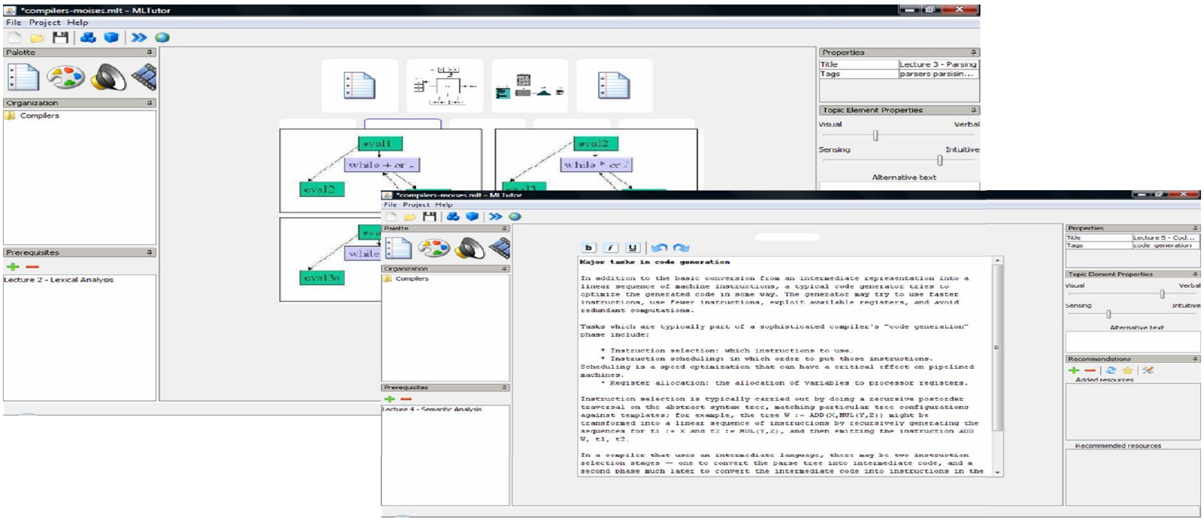
Participants in the evaluation of the authoring tool were two groups of 30 individuals mainly engineering students at the Instituto Tecnológico de Culiacán and teachers from the same institution, which participated in a workshop, set up expressly to use the tool. The workshop lasted four hours during which there were several tutors on basic computing, eolic energy, Mayan language and compilers. The compiler course structure is shown in Figure 3 using the interface to create courses of EDUCA. The tree-like structure shows two chapters, and six sections in addition to two questionnaires. For each chapter we added material or learning objects for different learning styles, pre-requisites and recommended resources from the Web (Figure 4). To assess user satisfaction of the software tool we applied a questionnaire of 20 questions to each participant at the end of the workshop, asking them to evaluate the software on the *Likert* scale according to their degree of agreement or disagreement.



Title	Compilers
Description	Compute Science 1...
Author	Alex Aiken - Moises...
Contact	mosorio@itculiacan...

Figure 3. *Educa main interface*

Figure 5 depicts the concentration of frequencies in the evaluation for five questions. As shown in the graph the results tell us that most of the group in the workshop "agree" or "strongly agree" with respect to usability of the interfaces, easiness to generate an intelligent tutoring system, time to learn and use the tool (The workshop lasted for 4 hours), time to produce a tutor, and the workshop organization.



Major tasks in code generation

In addition to the basic conversion from an intermediate representation into a linear sequence of machine instructions, a typical code generator tries to optimize the generated code in some way. The generator may try to use faster instructions, use fewer instructions, exploit available registers, and avoid redundant computations.

Tasks which are typically part of a sophisticated compiler's "code generation" phase include:

* Instruction selection: which instructions to use.
* Instruction scheduling: in which order to put those instructions. Scheduling is a speed optimization that can have a critical effect on pipelined machines.
* Register allocation: the allocation of variables to processor registers.

Instruction selection is typically carried out by doing a recursive postorder traversal on the abstract syntax tree, matching particular tree configurations against templates; for example, the tree W := ADD(X,MUL(Y,Z)) might be transformed into a linear sequence of instructions by recursively generating the sequences for t1 := X and t2 := MUL(Y,Z), and then emitting the instruction ADD W, t1, t2.

In a compiler that uses an intermediate language, there may be two instruction selection stages — one to convert the parse tree into intermediate code, and a second phase much later to convert the intermediate code into instructions in the

Figure 4. *Adding learning objects for different learning Styles*

Figure 6 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 a parsing topic (compilers). The second and third mobile phone gives us a trace of the students' learning styles (in three stages) along the course. This information is valuable in that it shows how students' learning styles are varying from the one is obtained with the ILSQ to the correct one which is obtained with help of the Kohonen Network.

4. RELATED WORK

To the best of our knowledge, there exists no authoring tools for ITS (Carmona *et al*, 2008; Murray *et al*, 2003) to produce tutoring systems in mobile platforms. Related work aims at identifying learning styles using different approaches like Bayesian networks, decision trees, or Hidden Markov Models. However, the learning styles are calculated based only on the ILSQ questionnaire and none of those works are authoring tools (Carmona *et al*, 2008; Graf *et al*, 2008; Limongelli *et al*, 2008). There are several author tools used to create mobile applications like *MyLearning* (Attewell, 2004), *zirada* (Zirada, 2009), Test Editor (Romero *et al*, 2006), or mediaBoard (Attewell, 2005). Some of them are more PocketPC's oriented and some are focused to quiz editing or game-based learning. None of those author tools however, have the ability of adapting to the user's learning style, nor have portability across different computer and operating system platforms.

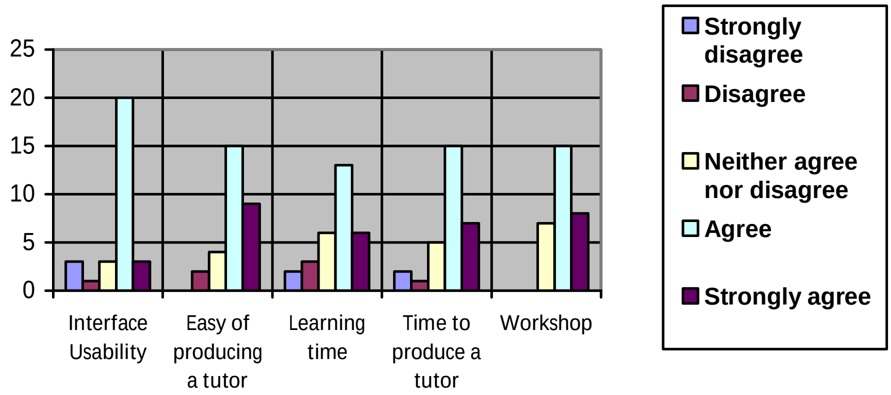
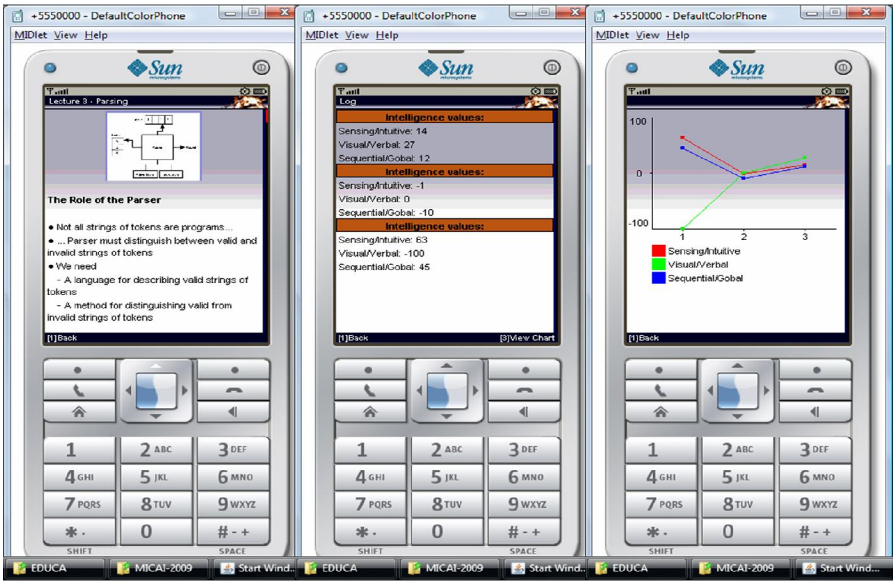


Figure 5. *Evaluation of the authoring tool*

5. CONCLUSIONS AND FUTURE WORK

This paper presents an approach for modeling Students' Learning Styles in Web 2.0 Collaborative Learning Systems using a Kohonen Network. The approach was implemented under an author tool that allows the production of personalized learning material to be used under collaborative and mobile learning environments. The software tool was implemented with Java version 1.6 and XML. The learning material produced with the tool is platform independent and it has been tested with different software emulators (NHAL Win32 Emulator, J2ME Wireless Toolkit and Sony Ericsson SDK), cell and smart phones (Sony Ericsson and Nokia). We present a test with one such mobile devices. Currently empirical studies are taking place to examine the students' reaction to the learning material produced using Educa. Future work should focus on continuing testing with official and complete material for course programs in public and private schools in Mexico. We store the complete authoring tool, in web address 201.120.128.222/educa, together with courses created in the workshop.



Lecture 3 - Parsing

The Role of the Parser

- Not all strings of tokens are programs...
- ... Parser must distinguish between valid and invalid strings of tokens
- We need
  - A language for describing valid strings of tokens
  - A method for distinguishing valid from invalid strings of tokens

[1]Back

Log

Intelligence values:
Sensing/Intuitive: 14
Visual/Verbal: 27
Sequential/Gobal: 12

Intelligence values:
Sensing/Intuitive: -1
Visual/Verbal: 0
Sequential/Gobal: -10

Intelligence values:
Sensing/Intuitive: 63
Visual/Verbal: -100
Sequential/Gobal: 45

[1]Back [3]View Chart

[1]Back

Figure 6. *The compiler course in a mobile phone*

AKNOWLEDGEMENTS

The work described in this paper is fully supported by a grant from the DGEST (Dirección General de Educación Superior Tecnológica) in México under the program “support and development of academic bodies” [Academic Body: Research in Software Engineering].

REFERENCES

Attewell, J. (2004). Mobile technologies and learning: A technology update and mlearning project summary. http://www.m-learning.org/reports.shtml [November 1, 2009].

Attewell, J. (2005). From Research and Development to Mobile Learning: Tools for Education and Training Providers and their Learners. fttp://www.mlearn.org.za/papers-full.html. [November 1, 2009].

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interactions, vol. 12(4), pp. 331--370, Kluwer Academic Publisher.

Carmona, C., Castillo, G., and Millán, E. (2008, July). Designing a Dynamic Bayesian Network for Modeling Student’s Learning Styles. Paper presented at the International Conference on Advanced Learning Technology, Santander, Spain.

Felder, R. M., and Silverman, L.K. (1988). Learning and Teaching Styles in Engineering Education. Engineering Education, (pp.674--681), vol. 78.

Felder, R. M., and Solomon, B. A. (2004). Index of Learning Styles Questionnaire. http://www.engr.ncsu.edu/learningstyles/ilsweb.html. [November 1, 2009].

Graf, S., Kinshuk, and Liu, T. (2008, July). Identifying Learning Styles in Learning Management Systems by Using Indications from Students' behavior. Paper presented at the International Conferenced on Advanced Learning Tecnology, Santander, Spain.

Hage, H., Aimeur, E. (2008, June). Harnessing Learner's Collective Intelligence: A Web 2.0 Approach to E-Learning. Paper presented at the International Conference on Intelligent Tutoring Systems, Montreal, Canada.

Kerly, A., and Bull, S. (2008, June). Paper presented at the International Conference of Intelligent Tutoring Systems, Montreal, Canada.

Kohonen, T. (2001), Self-Organizing Maps, 3rd ed., Springer, Berlin.

Limongelli, C., Sciarrone, F., and Vaste, J. (2008, July). LS-PLAN: An Effective Combination of Dynamic Courseware Generation and Learning Styles in Web-based Education. Paper presented at the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Hannover, Germany.

Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education, vol.10, 98—129.

Murray, T., Blessing, S., and Ainsworth, S. (2003). Authoring Tools for Advanced Technology Learning Environments. Kluwer Academic Publishers.

O'Reilly, T. (2005). What is Web 2.0. http://oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. [November 1, 2009].

Romero, C., Ventura, S., Hervás, C., and De Bra, P. (2006, June). An Authoring Tool for Building both Mobile Adaptable Tests and Web-based Adaptive or Classic Tests. Paper presented at the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Hannover, Germany.

Zatarain-Cabada, R., Barrón-Estrada, M.L., Sandoval, G., Osorio, M., Urías, E., and Reyes-García, Carlos A. (2008, October). Authoring Neuro-fuzzy Tutoring Systems for M and E-Learning. Paper presented at the Mexican International Conference on Artificial Intelligence, Hannover, Germany.

Zirada Mobile Publisher. (2009). The Premier Mobile Content Creation Tool. http://www.Zirada.com. [November 1, 2009].