Expert Systems with Applications 38 (2011) 9522–9529

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network

Ramón Zatarain Cabada [a,*], María Lucía Barrón Estrada [a], Carlos Alberto Reyes García [b]

[a] Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n col. Guadalupe, Culiacán, Sinaloa, CP 80220, Mexico
[b] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, CP 72840, Mexico

ARTICLE INFO

Keywords:
Intelligent tutoring systems
Cooperative/collaborative learning
Authoring tools and methods
Learning communities

ABSTRACT

This paper presents a Web 2.0 Learning Environment, for a systematic creation of adaptive and intelligent tutoring systems. Authoring contents is made by a community of users including teachers and students. The tutoring systems adapt the contents according to the best learning style using self-organizing maps (SOMs). The SOM was trained for classifying Felder–Silverman learning styles. The most important advantage of these unsupervised neural networks is that they do not require an external teacher for presenting a training set. The approach was implemented under an authoring tool that allows the production of personalized learning material to be used under collaborative and mobile learning environments. 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 authoring tool.

1. Introduction

Since the beginning of Intelligent Tutoring Systems (ITS) research, more than twenty authoring systems for ITS have been developed (Murray, Blessing, & Ainsworth, 2003). Those authoring tools can be classified according to what type of tutoring system they produce. For example the author tool SIMQUEST (de Jong et al., 1999) produces "simulation-based learning" systems, IRIS (Arruarte, Fernández, Ferrero, & Greer, 1997) creates "multiple knowledge types" systems and InterBook (Brusilovsky & Schwarz, 1998) 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 the simulation model, the learner interface, the instructional design, and the environment. In this list of tasks, the learners are never involved. In IRIS, the author produces an ITS by following two main phases: Specifying the general requirements of the tutoring system, and filling the learning contents of the ITS. On the other hand, 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 (Carmona, Bueno, EduardoGuzman, & Conejo, 2002), NetCoach (Weber, Kuhl, & Weibelzahl, 2001), and Ale (Specht, Kravcik, Klemke, Pesin, & Hüttenhain, 2002) differentiate only by the interface tools they operate. Some of them use special markup languages and some GUI interfaces. The author creates the learning environments and the learner read the learning contents.

In the last years 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). This kind of technology is now identified as *Social Software*. In the field of e-learning, a "tutor centered" concept is shifting to become more learner centered (coined as *e-learning* 2.0), where learners are also part of a community of authors and users of the learning resources (Hage & Aimeur, 2008). On the other hand, 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 the individual students (Kerly & Bull, 2008). Many approaches and implementations have been developed in recent years in order to model students' learning styles (Carmona, Castillo, & Millán, 2008; Graf, Kinshuk, & Liu, 2008). Most of those implementations use Bayesian Networks (Carmona et al., 2008), Linear Temporal Logic (Limongelli, Sciarrone, & Vaste, 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 (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, 1989). 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

* Corresponding author.
E-mail address: rzc777@hotmail.com (R.Z. Cabada).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.01.145

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.

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 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. The methodology

2.1. Building a tutoring system with Educa

The process of constructing a new tutoring system (an adaptive or intelligent system) consists of three main steps (Fig. 1). During Step 1 a tree structure of the adapted or intelligent tutoring system is designed by the main instructor(s) of the learning material. In the structure the designer specifies the course information (title, general description, author name, etc.), the unit names, and for each sub unit, the designer defines names and tags related to that sub unit, and all the prerequisites the student should get done. In the tree structure, the designer also inserts quizzes (multiple selection and choice). Quizzes are an important element to provide adaptation capabilities to the produced tutors.

In step 2 the tree structure is filled with domain contents (a knowledge base), and some other learning resources. At the beginning of the creation the instructor or teacher authors the tutoring system by inserting different learning objects like text fragments, images, audio/video clips, and by defining learning styles, prerequisites, tags and quizzes. At a later time, learning resources are added to the tutoring system by learners, who make recommendation of resources they find commonly in the web. Those resources can be found also in a special resource repository of Educa. After the author creates the **tutoring system**, she/he can save it and export it to a Mobile Learning format used to display tutoring systems on mobile devices (step 3). The saved/exported file will enclose three elements: a XML file corresponding to the **learning resources or contents,** a **predictive engine** for navigation purposes, and the **SOM Neural Network** for learning style classification. Another option to the output of Educa is to export the tutoring system to SCORM format. The benefit of this format is the availability of the material in any distance learning environment.

2.2. Educa architecture

Educa Architecture is composed by five main components or modules (Fig. 2): an authoring tool, two repositories (for resources and courses), an intelligent delivery Engine, and a recommendation engine.

The authors create initially a tutoring or adaptive system (a special type of knowledge base) using Educa authoring tool which runs in a desk computer. The tutoring system then is exported to a mobile computer (it can also be run in one wireless emulator like Sun Java Wireless Toolkit). Once the system is installed in a mobile device, a student can display the learning objects (LO) by using an Intelligent Delivery Engine (installed together with a neural network in the mobile). The engine displays the LOs in the mobile according to the learning style of the student and with the help of the neural network.

When the authors create the LOs, they have two different options: One choice, the traditional one, is importing LOs from different sources like SCORM, HTML, .doc or .PDF files or by editing directly the learning material by using a text editor of Educa. The second choice is more Web 2.0 oriented, and it consists of using a recommendation engine. This application employs two repositories: a resource repository, which is used for recommending new resources found in the Web by community members; a course repository which stores all the courses created with Educa. Both repositories are stored in a DBMS.

2.3. Creating the knowledge base

As we have mentioned before (Section 2.1), a course is created by first step designing a tree structure of the adapted or intelligent tutoring system, which represents the knowledge of the system. In order to designing and implementing a course knowledge representation, we apply some of the concepts related to Knowledge Space Theory (Doignon & Falmagne, 1999). This theory provides a sound foundation for structuring and representing the knowledge domain for personalized or adapted learning. It applies concepts from combinatory theory and we use it to model particular or personalized courses according to different learning styles. Educa enables the creation or construction and application of discipline-specific knowledge spaces (or structures). An adapted assessment based upon knowledge spaces and student prerequisites (and employing a Kohonen neural network for identifying learning styles), will derive a particular or personalized path of learning objects. Fig. 3 shows an example of the knowledge domain for a Compiler course.

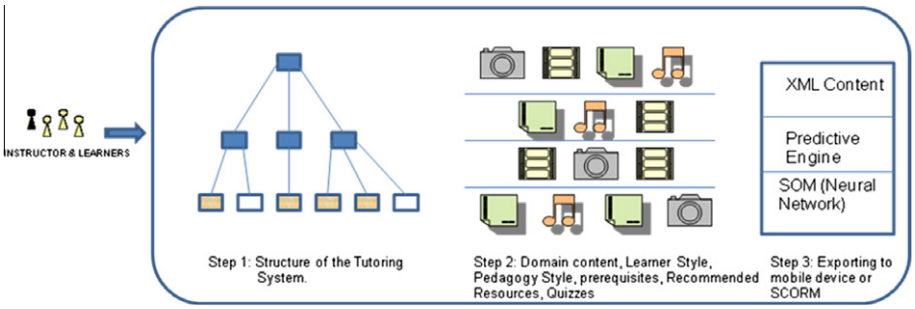

Fig. 1. Building a tutoring system with Educa.