Intelligent Learning Environments

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 106,** noviembre 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 106**, November 2015. Printing 500.

Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 106

Intelligent Learning Environments

Ramón Zatarain Cabada,
María Lucía Barrón Estrada,
María Yasmín Hernández Pérez (eds.)

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA
Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Editorial

Education and Artificial Intelligence have been combined to create a new area with its own fields of knowledge and technology application. This area involve working with diverse fields of expertize like pedagogy, machine learning, and now psychology. Traditional Intelligent Tutoring Systems (ITS) are capable to regulate student's learning on several levels. The potential value of this technology is obvious but they leave out important and modern issues of today´s learning process like a wide variety of learning settings, such as outside-of-school locations and outdoor environments. From this perspective, Intelligent Learning Environments give a support which combines the features of traditional ITS with modern and virtual learning environments.

In this volume we present seven research works in some of the most interesting fields of intelligent learning systems.

The papers were carefully chosen by the editorial board on the basis of three reviews by the members of the reviewing committee. The reviewers took into account the originality, scientific contribution to the field, soundness and technical quality of the papers.

We appreciate the work done by members of Mexican Society for Artificial Intelligence (Sociedad Mexicana de Inteligencia Artificial) and Instituto de Investigaciones Eléctricas (IIE) for their support during preparation of this volume.

Ramón Zatarain Cabada
María Lucía Barrón Estrada
María Yasmín Hernández Pérez

October 2015

Table of Contents

Page

Sequencing of Learning Objects based on SCORM Using cmi Elements and JavaScript

Miguel Sánchez-Brito[1], José Ruiz-Ascencio[1], Carlos Felipe García-Hernández[2]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Cuernavaca, Mexico

[2] Instituto de Investigaciones Eléctricas (IIE), Cuernavaca, Mexico

{miguelsb, josera}@cenidet.edu.mx, cfgarcia@iie.org.mx
http://www.cenidet.edu.mx

Abstract. In this research work a method for sequencing learning objects based on the SCORM standard using cmi tracking elements from this standard and javascript programming language, is presented. Advanced sequencing rules, dependent on student performance are achieved with SCORM 1.2, surpassing the present capabilities of learning objects in learning management systems. An analysis of a case study is also presented, showing that the configured code works correctly.

Keywords: SCORM, Moodle, cmi, learning object, LO, Captivate, JavaScript, sequencing, SCO, Multi SCORM, LMS, ADL, Reload Editor, on-line course.

1 Introduction

At present, more and more educational institutions have increased their attendance capacity, thanks to on-line education, that allows a student to carry out his/her school activities from any place with a computer connected to Internet, at any time. When choosing self-education, in most cases, on-line education requires Learning Objects (LO) with a proper pedagogical support, and attractiveness to students. Additionally, it is of great importance that the teaching material be adaptable to each student, based on his/her interactions in a specific activity of the teaching material. In other words, the learning process will be improved when using an LO that is really a challenge for the student. In this research work, a method for sequencing LOs based on "JavaScript" programming language, is presented, including the cmi tracking elements of SCORM standard (Shareable Content Object Reference Model, version 1.2), Captivate authoring tool from Adobe and the Learning Management System (LMS) Moodle (version 1.9.+) [4].

This paper is organized as follow: Section 2 provides related work on Shareable Content Object (SCO) sequencing and an application of pedagogical tools to an on-line course for e-learning. Section 3 presents the LO, SCORM standard, cmi and their capabilities in Moodle. Section 4 covers the capture of tracking elements in the LMS´s LO. Section 5 presents how to get the cmi values using javascript. Section 6 presents the resulting performance evaluation of code. Finally Section 7 and 8 cover the future work and conclusion for this research.

2 Related Work

In García-Hernández et al. authors explain in detail the way to achieve configuration of SCORM activities [5], involving sub-activities or several Shared Content Objects (SCO as named in SCORM standard, or learning objects containing teaching materials), following the guidelines proposed by ADL with Template 10 and Educative Model 3, and using Reload Editor software. Two forms of achieving sub-activities sequencing of a SCORM activity are described: applying learning objectives mapping for each activity (with the authoring tool) and setting ADL objectives with Reload in order to evaluate them, using shared global objectives and cumulative sub-activities (roll out). In Anbar et al. [6] authors explain an application of pedagogical tools to an on-line course for e-learning, including an example of a script for an activity, and a course development model.

3 LO, SCORM Standard, cmi and their Capabilities in Moodle

An LO is a teaching block configured with different multimedia elements (text, audio, images, animations, videos, etc.), aimed to motivate the learning process in a specific subject. Advanced Distributed Learning (ADL), a U.S.A. organization, is in charge of updating and publishing the SCORM Standard, the most recent version being the “2004 Fourth Edition”. SCORM is a standard that indicates how to develop teaching material (LO), for on-line learning (e-Learning) [2].

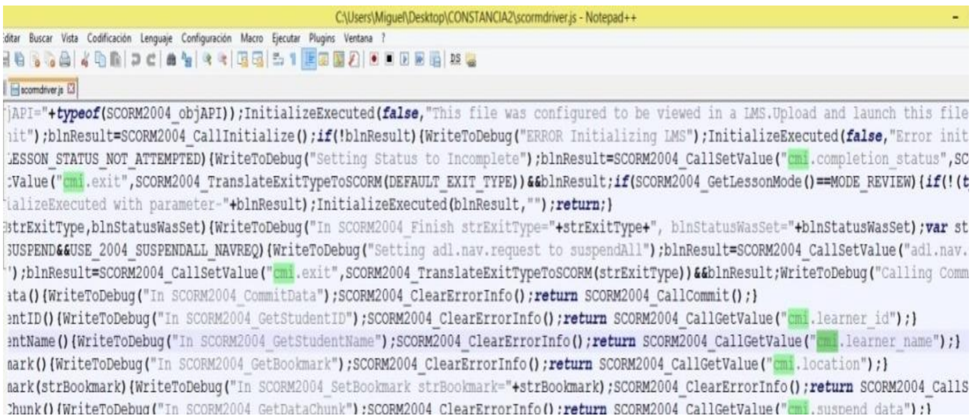


Fig.1. cmi elements of an LO.

Development of an LO based on SCORM is carried out with an authoring tool (Captivate, for this case study). Captivate is for editing text, inserting images, videos, animations, etc., and once the configuration of an LO is finished, it is published ("Save As" .zip extension file), in order to install it in LMS platforms. When publishing an LO, several code files are generated, that allow communication between an LO and an LMS platform. When using Captivate, a file called “scormdriver.js” is generated (among other files), which contains all the tracking elements or cmi elements of the LO. In figure 1, some cmi elements are shown.

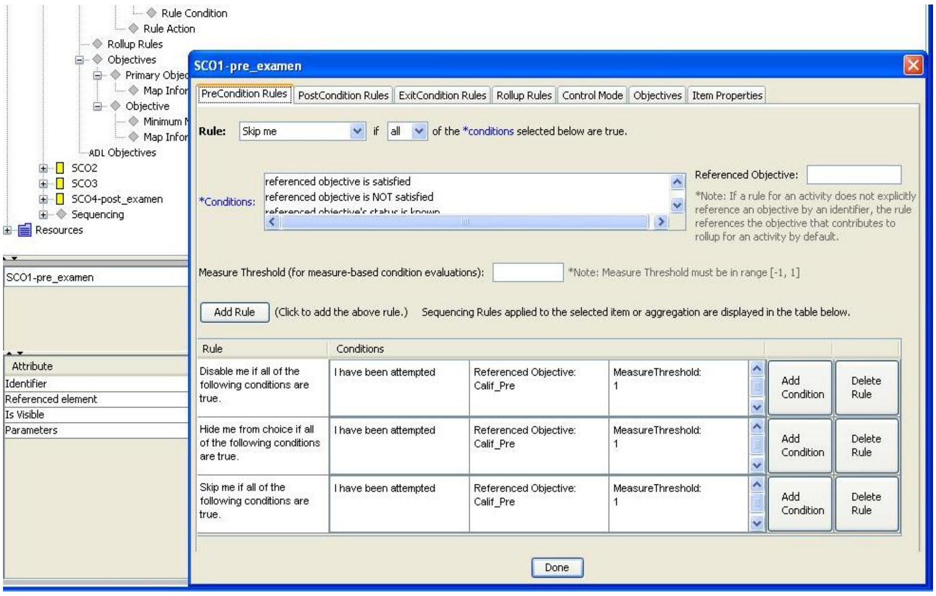


Fig.2. Configuration of advanced sequencing rules with ReLoad Editor is presented.

An LMS is a software system that carries out several activities required in an on-line school (user enrollment, activities, exams, internet links, students follow-up, grade book, etc.).One of the most relevant activities for the learning process, is the LO. However, at present most LMS (including Moodle) do not work correctly with an LO based on SCORM 2004, since they only support version 1.2 (previous to version 2004). Hence, this presents a great inconvenience for the learning process, because SCORM 2004´s capabilities [1, 3] allow presenting LOs with more than one activity or Multi SCORM (M-SCO), according to this standard. They also allow presenting it in an organized way that depends on objectives evaluation, completeness state, progress percentage and LO revision time. Hence, achieving advanced sequencing rules with SCORM 1.2 is the goal of this work.

In figure 2, a screenshot of the configuration process of advanced sequencing rules (SCORM 2004) with ReLoad Editor software (from ADL), is presented.

When loading the M-SCO sequence, the Moodle LMS established rules are skipped and the user will have access to all LO’s of the M-SCO, this presents a big problem if an LO is an assessment, because students may review any questions and then check other LOs to get the answer. This is due to the LMS platform supporting only SCORM version 1.2 and not having the necessary cmi elements to restrict the LOs of the M-SCO. How to sequence an M-SCO by means of the Reload Editor is detailed in [5].

4 Capture of tracking elements in the LMS´s LO

In order to obtain an advanced sequencing of the LOs in an M-SCO with SCORM version 1.2, it is proposed to use the cmi elements of each LO, with it a set of rules within an M-SCO which would be configured portable (being inside the .zip file or LO) and independent of each LMS platform, plus no modification of LMS would be required (with the installation of a system), since the M-SCO contains the necessary sequencing rules.

As mentioned above, the cmi elements are located in the scormdriver.js document, however to access them, it is necessary to call the function that contains them. For example, in scormdriver.js the "GetStudentName ()" function contains the element "cmi.learner_name". Figure 3 shows this function.

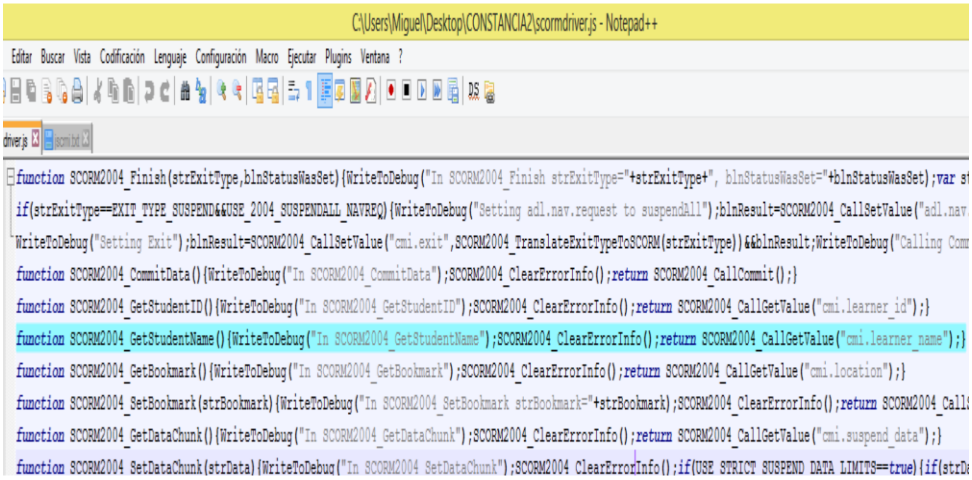


Fig.3. Function with which the student's name is obtained.

With the function mentioned, the student's name is obtained from the LMS, this variable could now be use (if necessary) in some LO, for example if required to grant a certificate to a user successfully passing a course. The "cmi.core.score.raw ()" function gets the user qualification obtained in any assessment.

5 How to Get the cmi Values Using Javascript

Authoring tools (Captivate in our case) have the option of adding a block of JavaScript code in a particular "slide". The inclusion of this type of code with the authoring tool Captivate can be appreciated in Figure 4.

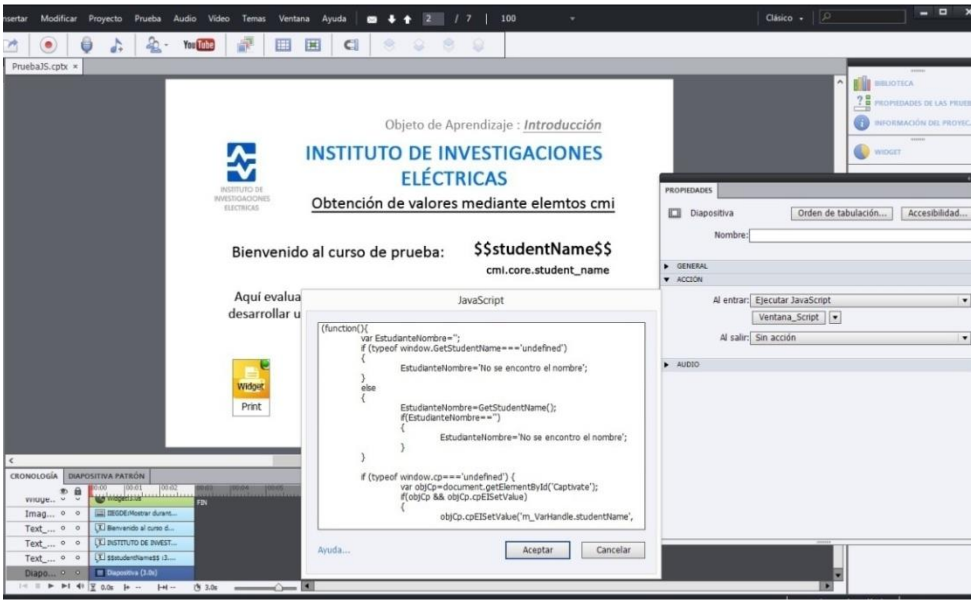


Fig.4. Adding JavaScript in Captivate.

The function providing the cmi values is shown in the following code block:

```
(function(){
        varEstudianteNombre='';
        if (typeofwindow.GetStudentName==='undefined')     {
                EstudianteNombre='No se encontro el nombre';
        }
        else      {
                EstudianteNombre=GetStudentName();
                if(EstudianteNombre==''){
                        EstudianteNombre='No se encontro el nombre';}
        }

        if (typeofwindow.cp==='undefined') {
                varobjCp=document.getElementById('Captivate');
                if(objCp&&objCp.cpEISetValue)   {
                        objCp.cpEISetValue('m_VarHandle.studentName',  EstudianteNombre);}
        }
        else {
                if(cp.vm && cp.vm.setVariableValue){
                        cp.vm.setVariableValue('studentName', EstudianteNombre);      }
        }
})();
```

The line "*EstudianteNombre=GetStudentName();*" is divided in two parts:

– ***GetStudentName***(): It is the most important line within the code as it allows us to access the functions contained in the cmi elements, this function is changing as we can get different cmi values.

– ***EstudianteNombre****:* It is a variable which is assigned the value obtained by the above function.

A complete list of cmi elements used in the SCORM standard is to be found in [2].

6 Results: Performance Evaluation of the Code

To verify the correct operation of the above code, by way of example an LO was developed in Captivate and mounted in the Moodle LMS of the Virtual Graduate Center of the Institute of Electrical Research (CPV-IIE).

Fig.5. Getting the name from the LMS platform.

Using "GetStudentName ()" we get the name of the user of the LMS platform, in Figure 5 we can see that the function performs its task properly.

One of the most important aspects for decision-making in any learning system is the score on any assessment. Because of that a test assessment which consists of only three questions was developed. Using *"cmi.core.score.raw ()"* we access the grade obtained by the user, and with the following line of code we developed a restriction for access to the next LO within the M-SCO:

```
if(EstudianteCalif>60)
        {
                window.open("Alerta.html","_self");
        }
```

The logic in the test is as follows:

If the user has two or more correct answers (it is equivalent to having a share of about 66% satisfaction, as there are 3 questions equivalent to 100%), the rule set for this example will launch a message warning the user he/she will be directed to new material.

Figure 6 shows the rule determined works correctly.

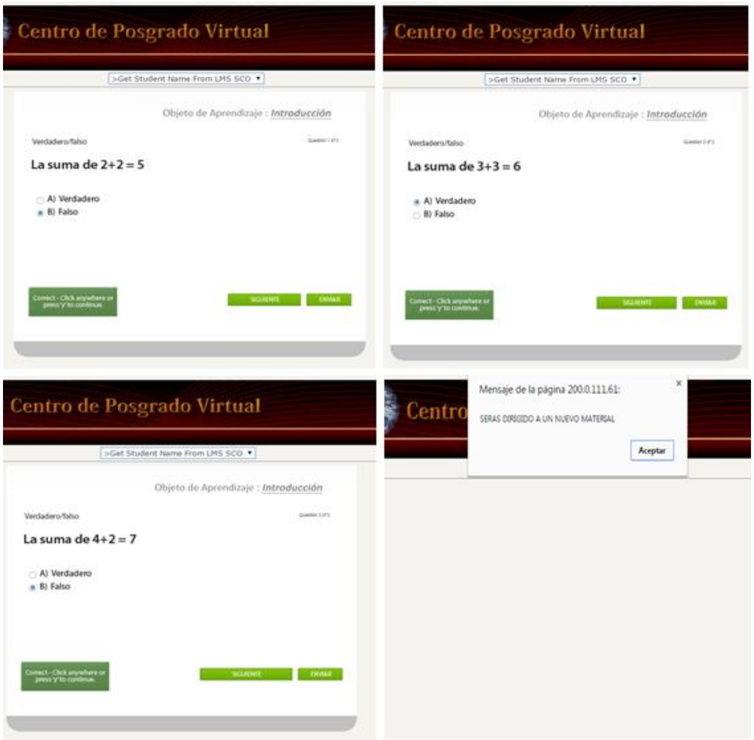


Fig.6. Assessment score detection by means of cmi elements.

Figure 7 shows the location of the file "Alerta.html" within the M-SCO, which contains the code above.

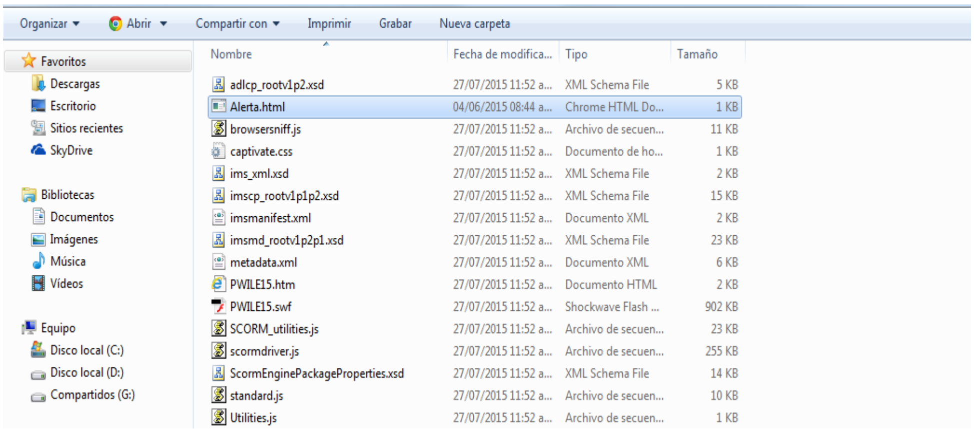


Fig.7. Location of JavaScript code to execute actions based on cmi elements of the SCORM standard.

With the latter, the use of the cmi elements for sequencing [6, 7] of the training material according to the performance of each student is demonstrated, thus being able to develop an advanced sequencing of (reusable) learning objects that meet the SCORM standard.

7 Conclusions

In the present research work a way to leverage the cmi elements of a learning object was exhibited in order to use it in any tutoring system. The tests presented here are evidence of the proper functioning of the code developed.

It is now possible to sequence learning objects in an advanced way and display them in an LMS, based on the SCORM standard for e-learning. This brings us one step closer to automating online courses delivered through an LMS, where the advanced sequencing is the last step of the intelligent tutoring.

8 Future Work

As future work we propose to develop an M-SCO containing several LOs and develop an intelligent system based on cmi and JavaScript elements, the latter would be very useful since it is practically a portable intelligent system when in the M-SCO with the capability of complying with the SCORM standard.

References

1. Miguel Sánchez-Brito, José Ruiz-Ascencio, Carlos Felipe García-Hernández: SCORM Cloud for an advanced sequencing of learning objects on LMS Moodle platform. Research in Computer Science, 87, pp. 19–26 (2014)
2. Advanced Distributed Learning, http://www.adlnet.org (2015)
3. Rustici Software, http://www.scorm.com (2015)
4. Moodle, http://www.moodle.org (2015)
5. C. F. García-Hernández, M. Sánchez-Brito, F. F. Jiménez-Fraustro: Configuración de la Plantilla 10 y Modelo Educativo 3 de ADL: Secuenciamiento con Remediación. In: X Congreso Internacional sobre Innovación y Desarrollo Tecnológico CIINDET, No. 620, Cuernavaca, Morelos, México, pp. 1–7 (2013)
6. Ahmad A. Anbar, Ahmad M. Al-Shishtawy, Marwa Al-Shandawely, Tamer A. Mostafa, Adham Bolbol, Ahmed Hammad, Saad Sunoallah, James Everett, Kutluk Özgüven: Applying Pedagogical Concepts in Online Course Development: Experiences from the Mediterranean Virtual University, pp. 1–11 (2005)
7. Guillermo Rodríguez Ortiz, Ricardo Molina González, Carlos Felipe García Hernández, Mayra Jazmín Mendoza Bahena: Metodología para la implantación de la capacitación a distancia. In: XII Congreso Internacional sobre Innovación y Desarrollo Tecnológico CIINDET, No. 7, Cuernavaca, Morelos, México, pp. 1–9 (2015)

An Architecture for Developing Educational Recommender Systems

Maritza Bustos-López, Raquel Vásquez-Ramírez, Giner Alor-Hernández

Instituto Tecnológico de Orizaba, Division of Research and Postgraduate Studies, Veracruz, Mexico

maritbustos@gmail.com, vz.rmz.raquel@gmail.com, galor@itorizaba.edu.mx

Abstract. The great amount of educational resources available on educational repositories enriches the learning process. However, it raises a new challenge: the need to provide support to the location of those resources that meet the needs, goals and preferences of each student. The location of useful educational resources to support the learning process is addressed by using recommender systems. Recommender systems are a tool to help student find information quickly and recommend new items of interest to the active student based on their preferences. In this paper, we propose a generic architecture for developing educational recommender systems independent of the type of recommendation generated. Also, we identify main features of an educational recommender system. These features are important components to achieve the objectives that have educational recommender systems in order to provide accurate information to students according to their preferences, user profile and learning objectives.

Keywords: Recommender systems, educational applications, collaborative filtering.

1 Introduction

Recommender systems are software tools and techniques providing suggestions for items to be of use to a user [1]. This kind of systems helps people to find items that the user is not aware, but that maybe of their interest. A recommender system assists in advertising tasks by automatically selecting the most appropriate items for each user according to his/her personal interests and preferences. Nowadays, many companies and Web sites implement recommender systems to study the preferences of users and adapt their products / information / services more appropriately to such interests in order to improve their results. In recent years, the interests in developing recommender systems has increased dramatically and have been proposed for different areas of knowledge as e-commerce, medicine, tourism, entertainment, education. In education, a recommender system imposes specific requirements and can take advantage of various types of knowledge in the referral process. For example, a recommender system can obtain the cognitive state of the student, which changes over time, and to carry out suggestions to change it. This would increase the level of customization in the long term. Another example, a recommender system can take advantage of a simple pedagogical rule as 'make difficult tasks easier ' or 'gradually reduce the amount of orientation'. Learning pathways and routes may represent sequences designed by professors from positive experiences in the classroom, or may correspond to the behavior of advanced students.

Recommender systems can be classified into five different categories depending on the technique employed to predict the utility of the items for the user, i.e., according to the recommendation technique [2]: (1) content-based recommender systems, (2) collaborative filtering recommender systems, (3) demographic recommender systems, (4) knowledge-based recommender systems, (5) hybrid recommender systems.

The implementation of techniques for the development of recommender systems is closely related to the type of information that will be used. A first source of information to keep in mind is the kind of elements that the system works. There are situations in which only an identifier of each element is known. For instance, in the case of the recommendation of educational resources, the title of the resource or resource content, the year in which the resource was developed, resource type, the author among other attributes are only known. This paper is organized as follows: Section 2 describes a review of previous related works. Section 3 presents the architectural design for educational recommender systems. The main features of Educational Recommender Systems are discussed in Section 4. Finally, future work and conclusions are presented in Section 5.

2 State of the Art

In the educational context, different recommender systems have been proposed in literature. A recommendation module of a programming tutoring system called Protus was developed in [3]. This module can be automatically adapted to the interests and knowledge levels of learners. Protus can recognize different patterns of learning style and learners' habits through testing the learning styles of learners and mining. Firstly, Protus processes the clusters based on different learning styles. Next, Protus analyzes the habits and the interests of the learners through mining the frequent sequences by the AprioriAll algorithm. Finally, Protus completes personalized recommendation of the learning content according to the ratings of these frequent sequences.

In [4] an educational collaborative filtering recommender agent was developed, with an integrated learning style finder. The agent produces two types of recommendations: suggestions and shortcuts for learning materials and learning tools, helping the learner to better navigate through educational resources.

A book recommendation system called PBRecS was developed in [5]. PBRecS is based on social interactions and personal interests to suggest books appealing to users. PBRecS relies on the friendships established on a social networking site, such as LibraryThing, to generate more personalized suggestions by including in the recommendations solely books that belong to a user's friends who share common interests with the user, in addition to apply word-correlation factors for partially matching book tags to disclose books similar in contents.

A new material recommender system framework based on sequential pattern mining and multidimensional attribute-based collaborative filtering (CF) was proposed in [6]. In the sequential pattern based approach, modified Apriori and PrefixSpan algorithms were implemented to discover latent patterns in accessing materials and use them for recommendation. Leaner Preference Tree (LPT) is introduced to take into account multidimensional-attribute of materials, and learners' rating and model dynamic and multi-preference of learners in the multidimensional attribute-based CF approach.

An online personalized English learning recommender system capable of providing ESL students with reading lessons that suit their different interests and therefore increase the motivation to learn was developed in [7]. The recommender system, using content-based analysis, collaborative filtering, and data mining techniques, analyzes real students' reading data and generates recommender scores, based on which to help select appropriate lessons for respective students. Its performance having been tracked over a period of one year, this recommender system has proved to be very useful in heightening ESL learners' motivation and interest in reading.

In [8], a personalized auxiliary material recommendation system was proposed based on the degree of difficulty of the auxiliary materials, individual learning styles, and the specific course topics. The proposal is based on several studies in which the effects of using Facebook were investigated in various aspects of education and a learning platform was used for the exchange of auxiliary materials.

A new multi agent learning system, called ISABEL was proposed in [9]. ISABEL provides each student, which is using a specific device, with a device agent able to autonomously monitor the student's behavior when accessing e-learning Web sites. Each site is associated, in its turn, with a teacher agent. When a student visits an e-learning site, the teacher agent collaborates with some tutor agents associated with the student, to provide him with useful recommendations.

An automatic personalization approach aiming to provide online automatic recommendations for active learners without requiring their explicit feedback was described in [10]. Recommended learning resources were computed based on the current learner's recent navigation history, as well as exploiting similarities and dissimilarities among learners' preferences and educational content. The proposed framework for building automatic recommendations in e-learning platforms is composed of two modules: (1) an off-line module which preprocesses data to build learner and content models, and (2) an online module which uses these models on-the-fly to recognize the students' needs and goals, and predict a recommendation list.

A personalized recommender system that used web mining techniques for recommending a student which (next) links to visit within an adaptable educational hypermedia system was described in [11]. A specific mining tool and a recommender engine were integrated in the AHA! system in order to help the teacher to carry out the whole web mining process.

In [12], a system that allows lecturers to define their best teaching strategies to be used in the context of a specific class is presented. The context is defined by: the specific characteristics of the subject being treated, the specific objectives that are expected to be achieved in the classroom session, the profile of the students on the course, the dominant characteristics of the teacher, and the classroom environment for each session, among others.

The system presented is the Recommendation System of Pedagogical Patterns (RSPP). A hybrid recommender system for learning materials based on their attributes to improve the accuracy and quality of recommendation was proposed in [13]. The system has two main modules: 1) explicit attribute-based recommender and 2) implicit attribute-based recommender. In the first module, weights of implicit or latent attributes of materials for learner are considered as chromosomes in a genetic algorithm then this algorithm optimizes the weights according to historical rating. Then, recommendation is generated by Nearest Neighborhood Algorithm (NNA) by using the optimized weight vectors implicit attributes that represent the opinions of learners. In the second module, preference matrix (PM) is introduced that can model the interests of learner based on explicit attributes of learning materials in a multidimensional information model. Then, a new similarity measure between PMs is introduced and recommendations are generated by NNA.

DELPHOS, a framework to assist users in the search for learning objects in repositories and which shows an example of application in engineering was proposed in [14]. LORs can be used in engineering not only for learning and training students, instructors and professionals but also for sharing knowledge about engineering problems and projects. The proposed approach is based on a weighted hybrid recommender that uses different filtering or recommendation criteria. The values of these weights can be assigned by the user him/herself or can be automatically calculated by DELPHOS in an adaptive and dynamic way.

In [15] a work-in-progress is presented with the aim of developing recommender system for personalization of activities in e-learning 2.0 environments. The main components of the proposed system are activity, student and group models, and recommender module. Activity model will be used for learning design representation and will include items that could be recommended to students: e-learning activities, possible collaborators, tools, and advices. To provide recommendations tailored to the student's and group's characteristics, an important component of the system will include student and group models. The recommender module, as third component of the system, will include original pedagogical rules together with the algorithms that adapt known recommendations techniques to the educational context.

Finally, E-learning resource recommendation was presented in [16], the project uses attribute of resources and learners and the sequential patterns of the learner's accessed resource in recommendation process. Learner Tree (LT) is introduced to take into account explicit multi-attribute of resources, time-variant multi-preference of learner and learners' rating matrix simultaneously. Implicit attributes are introduced and discovered using matrix factorization. BIDE algorithm also is used to discover sequential patterns of resource accessing for improving the recommendation quality.

In order to analyze more precisely the works described earlier, we present in table 1 a comparative analysis which summarizes relevant contributions of these related works.

These initiatives suffer from several drawbacks, such as: (a) Some works are based on the use of learning object as items to be recommended; (b) the multi-domain works use only one algorithm for the all domains; (c) these works are based only in the users rating for generating the recommendations. These deficiencies can be improved by: (a) a recommendation systems that implements a specified metric for each data type, utilizing the user and item information; (b) a system that automatically generates this kind of recommendation systems in an easy way providing a set of friendly user interfaces.

Table 1. Comparative analysis of proposals for educational recommender systems.

Research Work	Entity type	Objective	Algorithm	Recom mender System Type
A personalized English learning recommender system [8]	Unknown	Provide ESL students with reading lessons that suit their different interests and therefore increase the motivation to learn.	Clustering Association rules algorithm	Hybrid
Protus [3]	Learning objects	Estimate automatic recommendations to an active learner based on learning style and learning sequence.	AprioriAll Collaborative filtering	Collabora tive filtering
Personalized recommendation of learning material [6]	Learning objects	Propose a new material recommender system framework and relevant recommendation algorithms for e-learning environments.	Sequential pattern mining Apriori and PrefixSpan Nearest neighborhood	Hybrid
Hybrid attribute based recommender system for learning material [13]	Resourc e	Propose a hybrid recommender system for learning materials based on their attributes to improve the accuracy and quality of recommendation.	Nearest neighborhood Preference matrix Genetic Algorithm	Hybrid
DELPHOS [14]	Learning objects	Assist users in the search for learning objects in repositories and which shows an example of application in engineering	Unknown	Hybrid
A hybrid system of pedagogical pattern recommendations [12]	Unknown	Present a system that allows lecturers to define their best teaching strategies for use in the context of a specific class.	Singular value decomposition Nearest neighborhood	Hybrid
Application of implicit and explicit attribute for learning resource recommendation [16]	Resource	Propose a new resource recommender system framework for e-learning based on implicit and explicit collaborative filtering and sequential.	k-means Bi-Directional Extension based frequent closed sequence mining	Collabora tive filtering
U-Learn [4]	Learning objects	Help learners to accomplish their goals, offers suggestions of educational bibliographic materials and tools through a recommender agent based on learning style.	Unknown	Collabora tive filtering

3 Architecture of Educational Recommender Systems

In this section, we present a generic architecture for developing educational recommender systems. The architecture has a layered design in order to organize its components. This layered design allows scalability and easy maintenance because its tasks and responsibilities are distributed. The architecture of educational recommender systems is shown in Fig. 1. Each layer has a function explained as follows:

Presentation layer: This layer shows the end-user interface, allowing communication between the user and the systems. This layer is implemented by using HTML5, Java Script, and Cascading Style Sheets Level 3 (CSS3) for presenting information and allowing users to interact easily with the educational recommender systems.

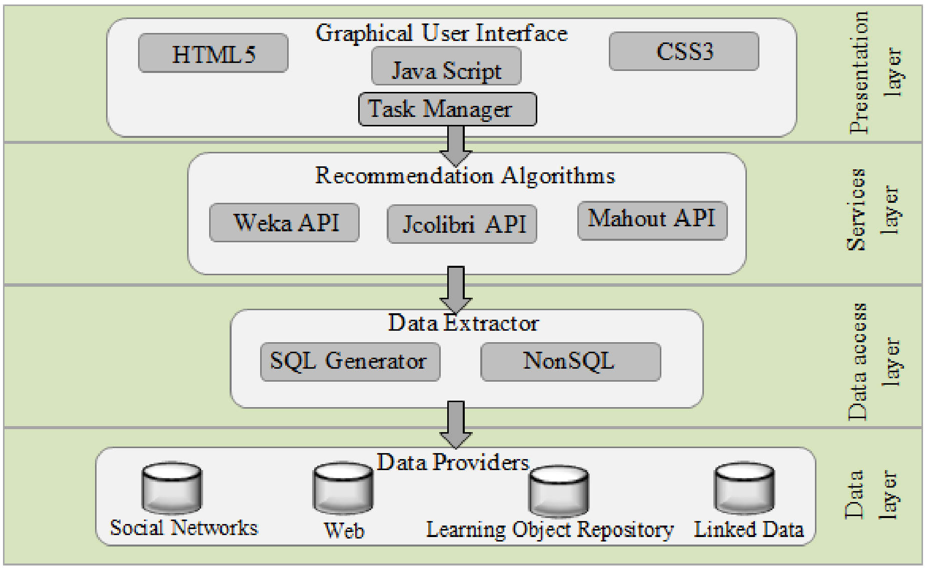


Fig. Educational Recommender Systems Architecture.

Services layer: This layer provides a set of APIs such as: 1) Weka which is a collection of machine learning algorithms for data mining tasks, 2) JColibri it is a java framework for building Case-based Reasoning (CBR) systems, and 3) Mahout API it is a library useful to build an environment for quickly creating scalable performant machine learning applications. These allow the generation of recommendations of educational resources. This layer has the recommendation algorithm that contains the metrics and algorithms for the generation of recommendations. The selection of the similarity metric is based over an analysis of several available metrics that are useful according to the data type.

Data access layer: A data extractor is located in this layer which maintains the data persistence, this layer is responsible of the proper execution of tasks such as insert, update, delete and query operations within educational recommender system's architecture. These tasks are encapsulated in this layer in order to provide security to the upper layers and avoid shortcuts on the data extractor.

Data layer: This layer stores information about the metadata educational resource provided by the educational recommender systems (video, books, and images, among others). Moreover this Data layer contains the different sources of information where educational resources may be located. As sources of information about educational resources for an educational recommender system, we can consider Social Networks as YouTube for video playing; SlideShare and Scribd to download slideshows, Picasa to visualize images. Other sources of information can be Learning Object Repositories. Some examples of these repositories are: MERLOT (Multimedia Educational Resource for Learning and Online Teaching), CAREO (Campus Alberta Repository of Educational Objects), SMETE (Science, Mathematics, Engineering and Technology Education) and others. At last, Linked Data can be considered where DBpedia is the main source of information for educational resources.

In this architecture, each component has a function which is explained as follows:

Graphical User Interface Component: This component represents the Graphical User Interface (GUI) which is responsible of handling the interaction between the user and recommender system. It is also a manager that captures user events, handles the task of validating all input data of the user. This component is responsible of handling user requests using the HTTP-based protocol.

Recommendation Algorithms Component: This component is responsible for the configuration of the algorithm implemented in the generated system. This component employs the WEKA API, JColibry API and Mahout API; it also includes several metrics for the calculation of similarity such as Pearson correlation, cosine measure and Euclidean distance.

Data extractor component: This component is responsible of retrieving all the information necessary for generating recommendation systems and its databases using SQL and NonSQL queries; therefore this component contains the SQL and NonSQL generator component, which allows generating the necessary SQL and NonSQL statements in order to create the system's persistence mechanism.

Data provider component: This component is responsible of providing persistence mechanism information required by the data extractor component. This information is located in Learning Object Repository, Social Networks, Web, Linked data.

4 Main features of Educational Recommender Systems

There are five main issues a recommender system must address. Firstly, a **knowledge acquisition technique** must be employed to gather information about the user from which a profile can be constructed. This knowledge is processed to provide the basis for an individual's user profile; it must thus be represented in a convenient way. There must be a knowledge source from which items can be recommended. Recommender systems allow information to be shared amongst users to enhance the overall recommendation performance; this shared information must be clearly defined. The final requirement is for an appropriate recommendation technique to be employed, allowing recommendations to be formulated for each of the users of the system.

Domain knowledge can also be shared, since it is normally programmed in and hence available to the system from the start. Categorizations of items can be used to provide order to a domain, and common sets of domain heuristics, potentially part of a knowledge base, can be useful when computing recommendations.

Table 2. Main features of educational recommender systems.

Feature	Description
Monitoring behaviour	An educational recommender system can observe the behavior of users based on their interaction with the system The system records this behavior to track student learning purposes instructional decision making and provide information to users on their progress.
Heuristics to infer information	An educational recommender system uses a set of rules that are used to infer information about users based on common preferences for decision making and finding information.
User feedback	An educational recommender system uses the process of user feedback to provide explicit information and thus confirm, add, overwrite, restructure information such as domain knowledge.
Filter rules	Recommender system provides filtering rules to users in search of accurate information that is related in the context of their learning.
User-created groups/categories	An educational recommender system allows users to define groups or categories in order to group users with learning styles and common characteristics (age, sex, skin color, socio-demographic characteristics among others) in the process of teaching and learning.
Item feedback	Item feedback is used by a recommender system to help users in making decisions for the selection of educational resources in the process of teaching and learning. These educational resources have been rated by other users. The user decides what, when, where, and what you want to study educational resources based on a feedback process.
Examples of items	Recommender system displays examples of items to form a collective training set.
Navigation history	A recommender system uses recorded navigation histories to help other users to find an optimal route.
Navigation trails	A recommender system has a navigation history that identifies the plan or path that a user continues in the learning process, such as a theme, a competition or a learning objective; and so recommend ways and shortcuts to educational resources.
Crawled web pages	An educational recommender system tracks Web pages to make this information as an ordered set of applications of pages visited by users. On this basis, browsing sessions are inferred in order to optimize and evaluate the recommendation of resources as well as propose alternative resources to users who share similar characteristics.
Internal database of items	An educational recommender system contains an internal database of items to generate the process of recommendation.
Similarity matching	Similarity function is used to find items matching a content-based profile.
Collaborative filtering	An educational recommender system uses the technique of similarity as a statistical function to find people with similar profiles, and then items liked by those people are recommended.

Profiles can be represented as a feature vector in a vector-space model. This is a standard representation and allows easy application of machine-learning techniques when formulating recommendations. For content-based recommendation the features in the vectors might be the word frequencies of interesting documents, while for collaborative filtering the features could be the keywords commonly used by users in their search queries. Navigation trails can be used to represent time-variant user behavior. If some initial knowledge engineering has been conducted there may also be knowledge about the users available to a profile.

The domain itself will contain sources of information to be recommended to the users. These could be from a database held by the recommender system, such as movie titles, or available dynamically via the web, such as links from the currently browsed page or web pages crawled from a web site. Systems can also rely on external events, such as incoming emails, to provide items for recommendation.

There is a wide variety of **recommendation techniques** employed today, with most techniques falling into three broad categories. Rule filters apply heuristics to rank items in order of potential interest. Machine-learning techniques employ similarity matching to rank items in order of interest. Collaborative filtering finds similar users and recommends items they have seen and liked before.

After analyzing about 60 papers related to develop recommender systems, we have identified the main features of a recommender system under an educational context. Table 2 describes these main features.

5 Conclusions and Future Work

In educational process, a recommendation system provides many benefits such as reducing the time spent searching for educational resources that meet the needs, preferences and objectives of a student. An educational recommender system incorporates different levels of customization to the user in order to formulate a query that will reflect what are his/her short- and long-term, that is, what he/she wants to learn in one session or future sessions of learning through an extraction process his/her profiles. For these reasons educational recommender systems are a perfect tool that provides personalized support to each student in their learning process. We have presented a literature review about educational recommender systems and we have designed (which has not been tested yet) a generic architecture for developing educational recommender systems. Also, we have identified the main features that an educational recommender system must have.

As future directions, we are considering developing the modules identified in the architecture to build an educational recommender system that meets characteristics presented in this paper. Also we are considering to include opinion mining, sentimental analysis and building user profiles to generate recommendations of educational resources for further customization and minimize problems locating educational resources scattered in different repositories.

Acknowledgments. This work was sponsored by the National Council of Science and Technology (CONACYT), the National Technology of Mexico (TecNM) and the Public Education Secretary (SEP) through PROMEP.

References

1. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (eds.), Recommender systems handbook, pp. 1–35, Springer (2011)

2. Luis Omar Colombo Mendoza, Rafael Valencia García, Alejandro Rodríguez González, Giner Alor Hernández, José Javier Samper Zapater: RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. Expert Systems with Applications,1202–1222 (2015)
3. Aleksandra Klasnja Milicevic, Boban Vesin, Mirjana Ivanovic, Zoran Budimac: E-Learning personalization based on hybrid recommendation strategy and learning style identification. Computers & Education, 56, 885–899 (2011)
4. Maria Iuliana Dascalu, Constanta Nicoleta Bodea, Alin Moldoveanu, Anca Mohora, Miltiadis Lytras, Patricia Ordoñez de Pablos: A recommender agent based on learning styles for better virtual collaborative learning experiences. Computers in Human Behavior, 243–253 (2015)
5. Maria Soledad Pera, Nicole Condie, Yiu-Kai Ng: Personalized Book Recommendations Created by Using Social Media Data. In: WISE 2010 Workshops, 6724, pp. 390–403 (2011)
6. Salehi Mojtaba, Isa Nakhai Kamalabadi, Mohammad Bagher Ghaznavi Ghoushchi. Personalized recommendation of learning material using sequential pattern mining and attribute based collaborative filtering. Educ Inf Technol (2012)
7. Mei Hua Hsu: A personalized English learning recommender system for ESL students. Expert Systems with Applications, 34, 683–688 (2008)
8. Hsua Chia Cheng, Chena Hsin Chin, Huangb Kuo Kuang, Huang Yueh Min: A personalized auxiliary material recommendation system based on learning style on Facebook applying an artificial bee colony algorithm. Computers and Mathematics with Applications, 64, 1506–1513 (2012)
9. Rosaci Domenico, Sarn Giuseppe M.L.: Efficient personalization of e-learning activities using a multi-device decentralized recommender system. Computational Intelligence, 26, 121–141 (2010)
10. Koutheaïr Khribi Mohamed, Jemni Mohamed, Nasraoui Olfa: Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. Educational Technology & Society, 30–42 (2009)
11. Cristóbal Romero, Sebastián Ventura, Jose Antonio Delgado, De Bra Paul: Personalized Links Recommendation Based on Data Mining in Adaptive Educational Hypermedia Systems. LNCS 4753, pp. 292–306 (2007)
12. Carlos Cobos, Orlando Rodriguez, Jarvein Rivera, et al.: A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes. Information Processing and Management, 607–625 (2013)
13. Salehi Mojtaba, Pourzaferani Mohammad, Amir Razavi Seyed: Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model. Egyptian Informatics Journal, 67–68 (2012)
14. Zapata, A., V.H. Menéndez, M.E. Prieto, C. Romero: A framework for recommendation in learning object repositories: An example of application in civil engineering. Advances in Engineering Software, 1–14 (2013)
15. Holenko Dlab, Martina, Natasa Hoic-Bozic: Recommender System for Web 2.0 Supported eLearning. In: IEEE Global Engineering Education Conference (EDUCON), pp. 953–956 (2014)
16. Salehi Mojtaba: Application of implicit and explicit attribute based collaborative filtering and BIDE for learning resource recommendation. Data & Knowledge Engineering, 87, 130–145 (2013)

Multicriteria Decision Making for Evaluation of e-Learning Tools

Eduardo Islas-Pérez, Yasmín Hernández-Pérez, Miguel Pérez-Ramírez, Carlos F. García-Hernández, Benjamín Zayas Pérez

Instituto de Investigaciones Eléctricas, Cuernavaca, Morelos 62490, Mexico

{eislas, myhp, mperez, cfgarcia, zayas}@iie.org.mx

Abstract. This document is in the required format. This work shows a benchmark of e-Learning tools including an approach for comparing them based on histogram specification concepts. The analysis is based on the definition of a set of criteria which are useful and desirable characteristics of learning management systems. The final results show the evaluation from different views including the approach based on their histograms. The evaluation of each e-Learning tool is based on the use of a three-dimensional model which organizes the criteria in three different axes according to their functionality inside the model, namely: Management, Technological and Instructional. With the application of the evaluation methodology we can assess the tools from different points of view. One of the main objectives of this work is to help users and developers of e-Learning tools to make good decisions about which tool have the best features for developing training and learning systems and for development and management of resources, courses and learning objects.

Keywords: e-Learning, evaluation methodology, learning management systems, multi-criteria decision making.

1 Introduction

The aim of this work is to present updated outcomes of a benchmarking of e-Learning technologies which is based on a proposed evaluation methodology within a three-dimensional model of criteria (3D model). The information, the evaluation methodology and the 3D model of criteria might provide useful information to e-Learning users and developers to make good decisions about which tool has or should have the best features for choosing or developing a management system of instructional resources such as courses and learning objects.

The proposed methodology in this work is very useful to evaluate the applicability of each learning tool from a global point of view as well as to establish the ranking of each learning tool in different standpoints.

Commercial and free LMSs, integrate different modules providing a complete learning tool. However, this integration increases the cost and complexity of each tool.

In this accomplishment we present evaluation results for eight Learning Management Systems.

Although the extant literature has many articles, books, internet services, and guides to evaluate LMS packages [1, 2]; they do not use the approach presented in this work, and where there is some similarity, the method is not described in detail as it is covered here. The evaluation methodology described can be used to evaluate other kind of items, using office tools and it can be adapted to evaluate other software products as Database Management Systems [3] or others. We have also applied the methodology to evaluate Virtual Reality development tools and even Virtual Reality equipment [4].

Nowadays an increasingly huge number of LMS packages are available; more than 165 are mentioned in [5] where it is also shown an evolution of several tools from open source platforms to commercial platforms.

The proposed methodology was used to update the evaluation of only five commercial platforms (Docebo, Joomla, Blackboard, IBM Social Learning and PeopleSoft) and three open source tools (Dokeos, Moodle and Sakai) since these LMS are still extensively used since our last evaluation described in [6]. Some of these tools (IBM Social Learning and PeopleSoft) are not LMS strictly but have some functionality related with e-Learning and e-Training, for instance they allow to load and publish instructional content, also allow asynchronous communication, etc. We believe that this evaluation although uneven for these tools might be useful for companies to make decisions about which tool fulfill their requirements to use in their e-Learning and e-Training activities [7].

The rest of this work is organized as follows: some related work is presented in section 2; section 3 describes the evaluation methodology; section 4 illustrates how the methodology was applied to evaluate different e-Learning tools from different perspectives including an approach for equating each tool based on histogram specification with respect to a sound e-Learning tool; finally section 5 provides some conclusions.

2 Related Work

The set of criteria in the 3D-model is based on [8], the main differences are how we apply them in evaluation: we assign several characteristics to each criterion and give different weights to each of them based on their relevance and we also group them in the 3-dimensional model in order to evaluate the tools from different perspectives such as instructional, management and technological.

Regarding evaluation methods, there are different approaches [9]; one of them distinguishes quantitative and qualitative methods. The former gives numerical results and the later use narrative or descriptive data rather than numbers [10].

Within the quantitative methods, one of the most used is the MultiCriteria Decision Making (MCDM) [11], [12]. At the core of the MCDM, a list of criteria must be defined; each criterion specifies a parameter to be evaluated, since they personalize a specific feature of the item under evaluation.

The MCDM are general purpose methods, in the sense that depending on the kind of items to be evaluated, they demand the definition of a specific set of criteria (or parameters) that an item in turn must accomplish with. The set of criteria in turn personalizes the methodology and at the same time makes it flexible enough to be applied in the evaluation of different kinds of items. However the sets of steps involved in a methodology might remain unaltered. That is, the criteria are different but the methodology is the same. This flexibility makes MCDM a powerful methodology with a large range of application. There is even a conference only in this topic and with this name MCDM [13]. This methodology has applications in Constructive Preference Learning in MCDA (Multiple Criteria Decision Aiding), Infrastructure Planning and Environmental Management, MCDA Models in Risk, Reliability and Maintenance Contexts, MCDM for smart and sustainable communities, among others [13].

We do have already used MCDM methodology to evaluated different kinds of items such as LMSs, Virtual Reality development tools, different types of hardware, etc., as long as we establish an appropriate set of criteria in each case.

3 Evaluation Methodology

In the next four subsections the complete methodology is described in several stages from criteria selection (i), until deployment of results and conclusions (vii)

3.1 Three-Dimensional Model Figures

The model relates the three most important aspects involved in personnel training and that constitutes the three axes of the 3D model, namely: Management (M), Technological (T) and Instructional (I) axes; these aspects allow three combinations between two axis (MT, MI, TI); and the combination of all of them (MTI). Accordingly this provides different viewpoints which allow evaluating each tool from seven different perspectives; these perspectives help to determine whether or not a tool fulfills the requirements from a Management, Technological or Instructional point of view.

The management dimension is related with administrative features of the tool and we can evaluate aspects such as: student tracking, curriculum management, statistics, etc. The instructional dimension deals with features related with instructional design, didactic planning, content production, instructor manual, student manual, and so on. Finally, the technological dimension involves attributes related to software and hardware tools used in the learning processes.

These three conceptual dimensions (axes) outline a 3D space and three planes, see Figure 1.

Each axis represents a set of attributes, so that the planes and the space represent different combinations of attributes of the axes involved. The attributes and combinations of attributes are shared by each criterion grouped in these planes, axes and the space. For instance, the criteria in the management-technological plane are helpful to determinate different management aspects made possible by technology. Thus, for example, instructors might be able to record delivery and review assignments of students online. This is possible because Learning Management Systems (LMS) are network platforms. Finally, the management-technological-instructional space indicates how the technology is being used to manage the learning process.

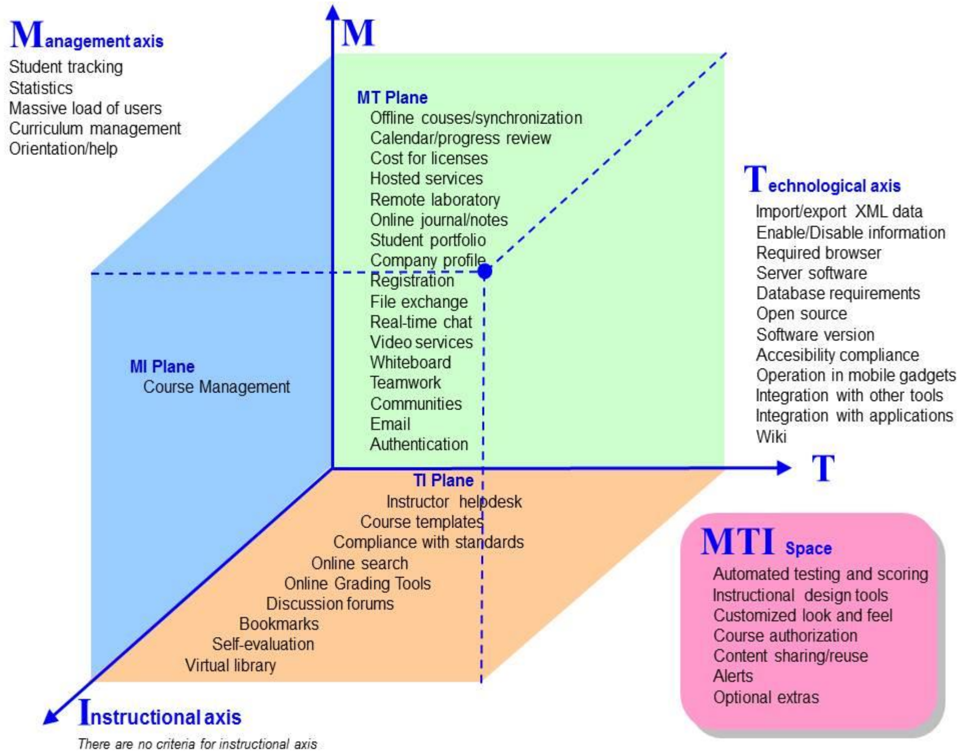


Fig. 1. Three-dimensional model (3D model) to evaluate modern learning and training systems.

3.2 Criteria and Weight Definition

The methodology is based on 51 criteria used to evaluate different technologies applied in modern training and learning systems. The same methodology was applied in our previous evaluation of e-Learning tools made in 2007 [6]. The criteria for e-Learning tools are grouped in the 3D model described above in accordance with their use and application in training and learning processes.

The methodology includes seven steps. This approach has been already applied successfully in [3, 6]:

i. Criteria selection. Once we know the kind of item to evaluate (LMSs in this case), in this step a group of criteria is defined. Each criterion will evaluate the degree of the item accomplishment of some requirement imposed by the user.
ii. Scales definition. A scale from 0 to 5 is assigned to every criterion; the scale will be useful to locate the degree of accomplishment of the criterion in turn by the tool in evaluation.
iii. Weight assignment. The importance of every criterion by assigning weights which range between 1 and 2 is defined. Thus, the number 2 is assigned to

those criteria which according to evaluator are more important for the company. On the other hand 1 is assigned, for instance, to needed criteria but being less important for the company. The purpose of the hierarchy is to make weighting easier; in theory, how it is structured should not affect the final weight assigned for each criterion [14, 15]. In experiments carried out in [14] non-hierarchical weights tend to be "flatter" (more equal), while hierarchical weights are "steeper" (have a greater variance).

iv. Selection of the items to evaluate. A set of specific items to be evaluated should be selected. This selection depends on the purpose of the items and on the evaluation itself. The criteria in turn represent the end users requirements.

3.3 Definition of Evaluation Methods

The evaluation study reported in [6], shows that three different [7, 14-17] Multi-Criteria Decision Making (MCDM) applied to the evaluation of LMSs was consistent, here is shown the additive value function and non-hierarchical weight assessment method (NWAM).

v. Analysis and evaluation of each LMS. Based on the criteria and the weight assignment, the tools were reviewed, analyzed and evaluated, grading them in accordance with the method NWAM:

MCDM: Additive value function and non-hierarchical weight assessment.

$$MAX\ V(A_{ij}) = \sum_{i=1}^{n} w_i v_i(x_{ij}), \qquad (1)$$

where:

x_{ij}	The value of criterion x_i for alternative A_j,
$v_i(x_{ij})$	A single criterion value function that converts the criterion into a measure of value or worth. These are often scaled from 0 to 1, with 1 being better. In this first method these values were not scaled.
w_i	Weight for criterion x_i, representing its relative importance. These are often normalized then: $\sum_{i=1}^{n} w_i = 1$ In this first method the weights were not normalized, instead they all were assigned with the same value of 1.
n	Number of criteria.
$MAX\ V(A_{ij})$	Higher values indicate a better tool.

3.4 Results

vi. Comparison of LMSs. In this step the results obtained are represented in different charts, considering each axis, plane, space, their histograms and the whole evaluation. These charts provide the means to compare them so that we can establish graphically strengths and weaknesses of the different tools under evaluation.

vii. Results obtained and conclusions. Based on all previous outcomes, finally we can draw some conclusions and the numerical outcomes will provide support to the decision making. Program listings or program commands in the text are normally set in typewriter font, e.g., CMTT10 or Courier.

4 Application of the Evaluation Methodology

Here is shown how the methodology is used to evaluate different LMSs.

4.1 Criteria Definition and Value Assignment

In Figure 1 the whole set of criteria are shown. Some criteria were taken from [8] and were grouped for each dimension (the criteria that involve only one dimension), in each plane (two dimensions) or in space (three dimensions).

As we stated in [6] "Although subjective, it is worth clarifying that this grouping is based on our experience and the criteria could be grouped in a completely different way, for instance some criteria can be included in a plane or in the space. For instance student tracking was classified as M (because student tracking usually is a Management activity) but it could have been classified in the MT plane (because this kind of management is achieved using LMS technology). It depends on the significance for users and developers on different aspects that a criterion might involve". Nevertheless the overall score of the evaluation of an LMS remains unchanged, no matter where the criteria are located, in an axe, in a plane or in the space.

That is to say, with the obtained results in this and our previous work [6] and since each tool is assigned the total sum of values of all criteria we can establish that axes, planes and space are going to provide us with detailed different approaches or views of the evaluations but the overall evaluation is constant.

In this work, only one criterion is described in detail, as well as its value assignment (Table 1). The rest of the criteria were analyzed in the same way and are described in detail in [6].

Likewise in [8] additionally to the evaluation, other activities involved in the analysis of modern training systems can be integrated to the 3D model or complementing it, for example:

i. A cost - benefit analysis might be needed so that a company would be able to make decision on purchasing, development or using an e-Learning tool. These in turns might involve time and technical support considerations.
ii. Correlate the company competences and the solution of specific problems that the enterprise faces.
iii. Take into account practical guidelines to optimize the use of technologies for instructional purposes.
iv. Etc.

Table 1. Student tracking criterion: Value assignment.

Features	
1. Only the tracking of exams can be carried out. 2. Every element can be selected to carry out the student's tracking (homework, tests, essays, final exam, projects, etc.) 3. Emissions of reports of every element in the course. 4. Different reports can be selected. 5. The reports can be configured to present one or several elements at the same time.	
Scale	**Description**
0	Student tracking is not supported.
1	The tool has one of the features above
2	The tool has two of the features above
3	The tool has three of the features above
4	The tool has four of the features above
5	The tool has five of the features above

4.2 LMSs Evaluation Results

The following subsections show an update of the results published previously in [6]. In this actualization are depicted the evaluations of the systems: Blackboard, Docebo, Dokeos, IBM Social Learning, Joomla, Moodle, PeopleSoft and Sakai. We chose these tools trying to include the most popular LMSs and learning tools. However they will be helpful to illustrate the use of the 3D model as an evaluation methodology.

4.2.1 Results Obtained for e-Learning Systems from Different Perspectives

The evaluation was carried out by assessing the degree of fulfilment of features of each criterion. Some e-Learning tool was tested and evaluated in collaboration with software providers where each feature of each criterion was reviewed. The results for all perspectives except for the Instructional axis (because it does not have any criteria associated in our approach) are presented in a graphical way in Figures 2, 3, 4, 5, 6 and 7.

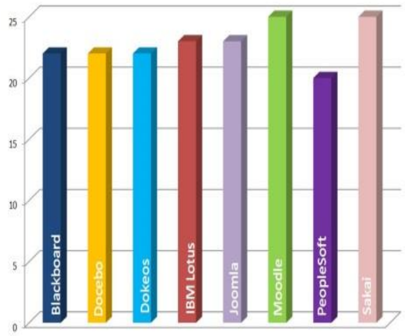


Fig. 2. Management axis.

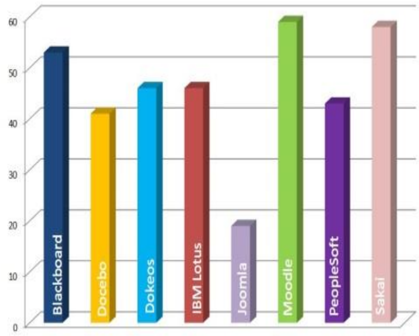


Fig. 3. Technological axis.

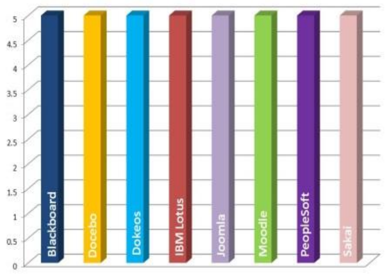


Fig. 4. Management-Instructional plane.

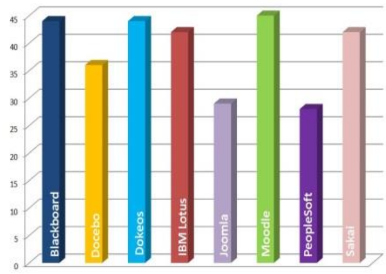


Fig. 5. Technological-Instructional plane.

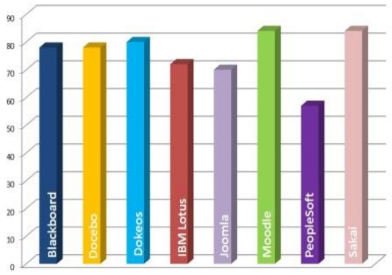


Fig. 6. Management-Technological plane.

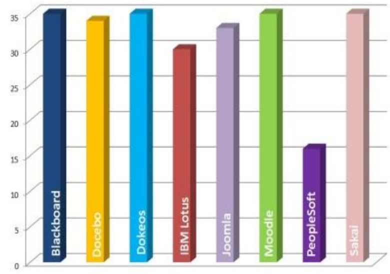


Fig. 7. Three-dimensional space (MTI).

4.2.2 Outcomes from Applying the MCDM

The results for the first MCDM method are depicted in Figure 8, which shows the ranking and global results for each software tool. These global results include all the criteria considered applying the additive value function without scaling the value function $V_i(X_{ij})$ and using non-hierarchical weight assessment. In this method, the best evaluated tool was Moodle followed by Sakai.

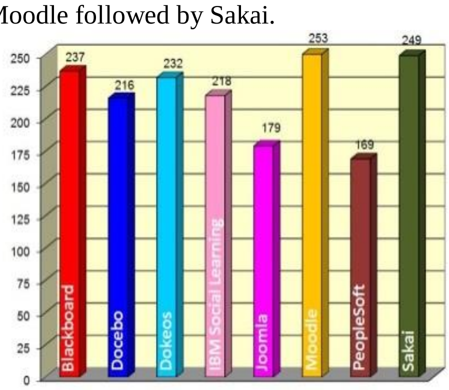


Fig. 8. Total sum of values using the MCDM.

Considering the results obtained in our previous work [6] and corroborating the grades of this actualization we can compare the methods and rules in terms of their ease of use, appropriateness and validity as it was stated in [6, 11, 12].

Ease of Use and Appropriateness.

Once the set of criteria and the weights for each criterion have been defined, to follow the MCDM method described above does not represent a problem. At most it will demand some time depending on the number of criteria. In [6] was shown that no matter what MDCM is used to evaluate, the outcomes are consistent.

4.2.3 Comparison of Results Using Histogram Matching and Histogram Specification

In the digital image processing area there is the concept of histogram equalization to produce an output image that has a uniform histogram. This output image features a good quality and worthy contrast. Also in this area the concept of histogram specification is useful to be able to specify the shape of the image histogram that we wish to accomplish. The method used to generate a desired output that has a specified histogram is called histogram matching or histogram specification [18].

In this section we use these concepts and present the obtained histogram for each e-Learning tool taking into account all the criteria (see Fig. 1). See the assessments shown in Figure 9.

The histogram allows having a general view about the degree of accomplishment of each criterion by the set of LMSs evaluated. Thus, we can see that *Remote laboratory* is the lowest accomplishment criteria by all the LMS evaluated. On the other hand we can observe that criteria such as *Required browser*, *Discussion forums*, *Authentication*, etc. are well accomplished by all LMSs evaluated.

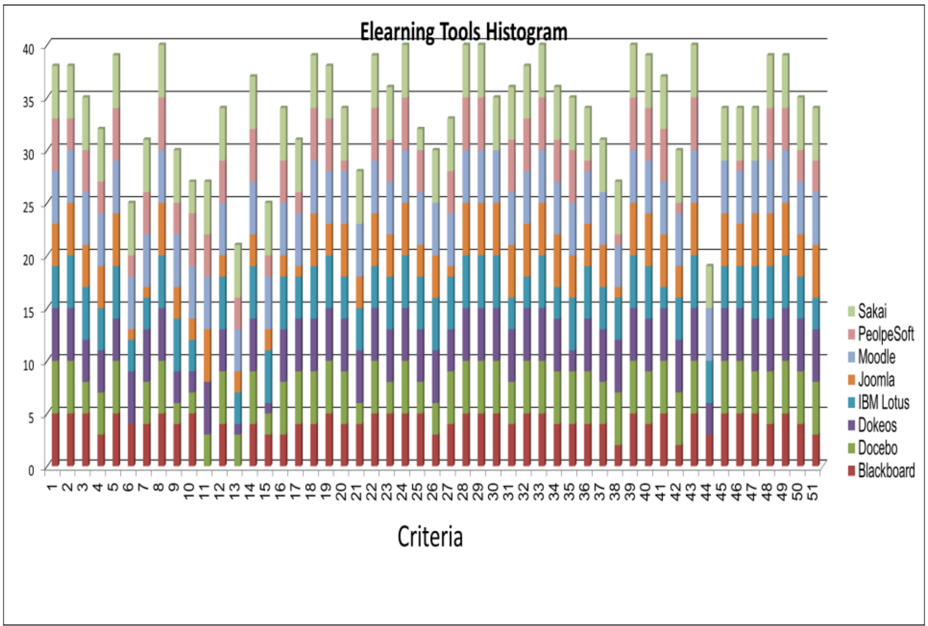


Fig. 9. E-Learning tools histogram, the numbered criteria are shown in Table 2.

Table 2. Set of criteria (based on [8])

No	Criterion	No	Criterion	No	Criterion
1	Student Tracking	18	Course Management	35	Real-Time Chat
2	Statistics	19	Instructor Helpdesk	36	Video Services
3	Massive load of users	20	Course Templates	37	Whiteboard
4	Curriculum Management	21	Compliance with Standards	38	Offline Courses/Synchronization
5	Orientation/Help	22	Online Search	39	Teamwork
6	Import/Export XML data	23	Online Grading Tools	40	Communities
7	Enable/Disable information	24	Discussion Forums	41	Student Portfolio
8	Required Browser	25	Bookmarks	42	Cost of licenses
9	Server Software	26	Self-evaluation	43	Company Profile
10	Database Requirements	27	Virtual Library	44	Remote Laboratory
11	Open Source	28	Calendar/Progress Review	45	Automated Testing and Scoring
12	Software Version	29	Authentication	46	Instructional Design Tools
13	Accessibility Compliance	30	Hosted Services	47	Customized Look and Feel
14	Operation in Mobile Gadgets	31	Registration	48	Course Authorization
15	Integration with other Tools	32	File Exchange	49	Content Sharing/Reuse
16	Integration with applications	33	Email	50	Alerts
17	Wiki	34	On-line Journal/Notes	51	Optional Extras

5 Conclusions

In the application of MCDM methods to make a decision based on the results, Hobbs and Meier [9] recommend to apply more than one approach because different methods offer different results to compare. In evaluating the results of different methods, the potential for biases should be kept in mind. The extra effort is not large and the potential benefits, in terms of enhanced confidence and a more reliable evaluation process, are worth. However the results shown in our previous work [6] and in this accomplishment deploy the same ranking of choices it does not matter the method used as opposed in [13]. The model can be used to analyze a broad variety of different e-Learning technologies.

The main benefits obtained with the evaluation of several e-Learning tools from a general perspective and from different points of view including the approach for equating tools with respect to a sound software tool are: (1) personnel related in evaluating and selecting an appropriate tool is now informed about the differences and accomplishment of each tool and (2) e-Learning firms can identify the opportunity areas and features where they can improve their tools. The decision for choosing an e-Learning tool can be made taking into account: management, technological and instructional characteristics. Furthermore, decision makers can make up an action plan and choose the best path to follow in order to integrate this technology into their learning and training processes.

References

1. Bill Brandon: 311 Tips for the Successful Management of an LMS or LCMS. 1st ed. USA: The e-Learning Guild, 42 p. (2006)
2. WCET Learn. A Guide for Planning Communications during LMS Selection, Implementation and Beyond. [Online] (2010)
3. Eduardo Islas, Eric Zabre, Miguel Pérez: Evaluación de herramientas de hardware y software para el desarrollo de aplicaciones de realidad virtual. Bulletin Electrical Research Institute, 28:61–67 (2004)
4. Miguel Pérez, Eric Zabre and Eduardo Islas: Prospectiva y ruta tecnológica para el uso de la tecnología de realidad virtual en los procesos de la CFE. Instituto de Investigaciones Eléctricas, Cuernavaca, México (2004)
5. e-Learning Industry. Learning Management Systems Comparison Checklist of Features. [Online] Available from: http://e-Learningindustry.com/learning-management-systems-comparison-checklist-of-features [Accessed: 02/09/2015] (2013)
6. Eduardo Islas, Miguel Pérez, Guillermo Rodríguez, Israel Paredes, Ivonne Ávila, Miguel Mendoza: E-Learning Tools Evaluation and Roadmap Development for an Electrical Utility. Journal of Theoretical and Applied Electronic Commerce Research, 2(1):63–75 (2007)
7. William Horton, Katherine Horton: E-Learning Tools and Technologies: A consumer's guide for trainers, teachers, educators, and instructional designers. 1st ed., Indianapolis, USA, Wiley Publishing, 592 p. (2003)
8. Edutools: CMS Home, Edutools. [Online], Available: http://www.edutools.info/course/ (2013)
9. Aravossis Konstantinos, Koutsiana Efrosini: Program Evaluation Methodologies. A comparative Assestment. Discussion paper series, 9(17): 387–404 (2003)
10. APCRC: Evaluation methodology. APCRC, Bristol (2015)
11. B. F. Hobbs, P. M. Meier: Multicriteria Methods for Resource Planning: An experimental comparison. IEEE Transactions on Power Systems, 9(4): 1811–1817 (November 1994)
12. Benjamin F. Hobbs, Vira Chankong, Wael Hamadeh, Eugene Z. Stakhiv: Does Choice of Multicriteria Method Matter? An experiment in Water Resources Planning. Water Resources Research, 28(7):1767–1779, DOI: 10.1029/92WR00712 (1992)
13. MCDM: 23rd International Conference on Multiple Criteria Decision Making MCDM 2015 - Bridging Disciplines, http://www2.hsu-hh.de/logistik/MCDM-2015/invitedsessions.html
14. William G. Stillwell, Detlof von Winterfeldt, Richard S. John: Comparing Hierarchical and Nonhierarchical Weighting Methods for Eliciting Multiattribute Value Models. Management Science, 33(4):442–450 (1987)
15. Michelle L. Bell, Benjamin F. Hobbs, Emily M. Elliott, Hugh Ellis, Zachary Robinson: An evaluation of multi-criteria methods in integrated assessment of climate policy. Journal of Multi-Criteria Decision Analysis, 10(5):229–256, DOI: 10.1002/mcda.305 (2002)
16. Vira Chankong, Yacov Y Haimes: Multiobjective Decisión Making: Theory and Methods. Amsterdam, North-Holland, 432 p. (1983)
17. Theodor Stewart: A Critical Survey on the Status of Multiple Criteria Decision Making Theory and Practice. The International Journal of Management Science, 20(5-6):569–586. DOI: 10.1016/0305-0483(92)90003-P (1992)
18. Rafael C. Gonzalez, Richard E. Woods: Digital Image Processing. 3rd ed., New Jersey, USA, Pearson, Prentice Hall, 954 p. (2008)
19. M. Baldonado, C.-C.K. Chang, L. Gravano, A. Paepcke: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr., 1:108–121 (1997)

Affective Environment for Java Programming Using Facial and EEG Recognition

María Lucía Barrón-Estrada, Ramón Zatarain-Cabada, Claudia Guadalupe Aispuro-Gallegos, Catalina de la Luz Sosa-Ochoa, Mario Lindor-Valdez

Instituto Tecnológico de Culiacán, Culiacán, Sinaloa,
Mexico

{lbarron, rzatarain, m03171007, m07170739, m05170485}@itculiacan.edu.mx

Abstract. We have developed an affective and intelligent learning environment that helps students to improve their Java programming skills. This environment evaluates cognitive and affective aspects of students in order to define the level of difficulty of the exercises that are more suitable for the them in its current condition. The cognitive aspects are: the number of mistakes, the difficulty level of the current exercise and the time spent in the solution. The affective aspects are: the acquired emotion from a facial expression and the acquired valence from electroencephalogram signals. This environment also uses a neural network for face recognition of basic emotions, a support vector machine to define the valence of emotion and a fuzzy inference engine to evaluate the cognitive and affective aspects.

Keywords: Intelligent learning environment, affective detection, affective computing, face recognition, EEG recognition.

1 Introduction

Traditional learning environments do not provide an individualized learning model; so, it is common having a classroom group of more than 30 members per teacher. This is a disadvantage because individualized instruction is considered the most effective way of teaching [1]. Also in these environments the emotional state of the student during the learning process is usually ignored. A student can easily fall into boredom if what is taught is not a challenge; this can lead to student to consider teaching as uninspiring and therefore show no motivation.

Another case may be that the student is intensely frustrated in trying to understand a topic. If he is sufficiently persistent there is a possibility that he may achieve success, but there is also the possibility that he could desist in his attempts due to the low sense of achievement gained by his effort.

In these scenarios, the work of Woolf et al. [2] shows different types of interventions that a tutor could carry out to consider the cognitive and affective aspect of the student. In the first case, for example, if a student is showing he/she masters a subject but he/she feels bored then it is recommended to increase the difficulty of the exercises. In the second case, if a student fails to progress and is stressed, is recommended that the tutor provide an alternative to the student: changing the exercise or taking a small break.

There have been numerous investigations of the relationship between affection and learning that have demonstrated that positive affective states as concentration, enthusiasm, and joy may lead to better learning [3], and that boredom is associated with poor learning and behavioral problems [4].

Because of the importance of affect in the learning process, detection of affection is a key component in the development of Intelligent Learning Environments (ILE) that are capable of responding to the affective needs of the student.

On the other hand, one of the main problems a computer science student has to face is to learn a programming language. In this kind of courses the percent of failure increases considerably [5].

This paper presents the implementation of an ILE that consider the student’s affective and cognitive needs to provide individualized instruction to students learning the Java programming language.

This paper is organized as follows: Related Works with affect recognition is presented in Section 2. Implementation of the ILE is described in Section 3. Section 4 describes the performance of the ILE. Conclusions and future work are given in Section 5.

2 Related Works

In this section, we present research works in the field of affective or intelligent tutors.

The Intelligent Tutoring System **Autotutor** [6] helps students to learn Newtonian physics, computer literacy, and critical thinking topics; this system simulates a human tutor by holding a conversation with the learner in natural language. The tutoring occurs in the form of an ongoing conversation, with human input presented using either voice or free text input. To handle this input, the system uses computational linguistic algorithms.

The distinguishing feature of ILE Java Affective with Autotutor is that it uses facial expressions and EEG signals for the emotion recognition.

HelpTutor [7] is an intelligent tutor for seeking help. This system was integrated into Geometry Cognitive Tutor, an intelligent tutor for learning basic geometry that provides step by step guidance to resolve complex problems, estimates the level of knowledge of the student and provides feedback and clues for each of the steps in solving the exercises.

Additionally, with the help of HelpTutor, the system provides feedback cognitive goal enabling students to improve their ability to ask for help.

The distinguishing feature of ILE Java Affective with HelpTutor is that it considers the emotional dimension of the student as well as the cognitive through the emotion recognition from photos and EEG signals of the student. The system makes use of them to determine the level of difficulty of exercises in order to try to avoid negative emotions within the context of learning (boredom, constant frustration) that can result in poor performance.

Gaze Tutor [8] is an Intelligent Tutoring System designed to tutor students on high school biology topics (e.g., cellular respiration, mitosis, ecological succession). The tutor uses an eye tracker to monitor a student’s gaze patterns and identify when the student is bored, disengaged, or is zoning out. The tutor then attempts to reengage the student with dialog moves that direct the student to reorient his or her attention patterns towards the animated pedagogical agent embodying the tutor.

The distinctive feature of ILE Java Affective with Gaze Tutor is that considers cognitive and affective aspects of students. The ILE uses facial expressions and EEG signals for the emotion recognition.

Java Sensei is an Intelligent Learning Environment with affective management for Java, designed to help students of programming to strengthen different areas of knowledge on Java in a web environment. The system evaluates aspects such as cognitive and emotional state of the student to take intervention strategies that made the pedagogical agent and adaptability processes performed by a set of recommendations [9]. ILE Java Affective considers more cognitive elements than Java Sensei and also uses EEG signals for the emotion recognition.

3 Implementation of ILE Affective Java

In this section we present the main architecture of the ILE. Fig. ***1*** shows the three relaxed or flexible layers of our architecture. The relaxed layers are less restrictive about the relationships between layers. The presentation layer has two components. The Intelligent-Affective layer has five components. The data layer has three components. Next, we explain the operation of each component.

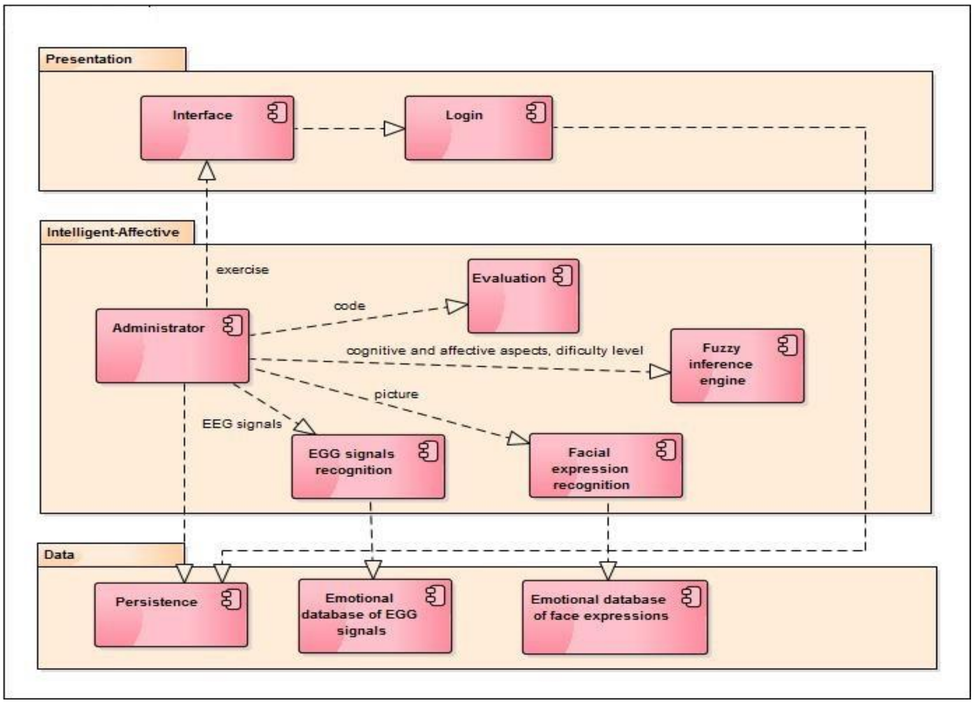


Fig. 1. Architecture of ILE.

Administrator: It is responsible for coordinating the tasks to be performed by the tutor and communicate with other components.

Evaluation: It is responsible for evaluating the exercise solved by the student and report any errors encountered.

Facial expression recognition: Identifies the student's emotion (happiness, anger, sadness, surprise and neutral) at the time of taking a picture of the student face.

EEG signal recognition: Identifies the emotion valence (positive, negative, neutral) of student through the EEG signals recorded during the resolution of the exercise. The ILE Affective Java works with the Brain-Computer Interface (BCI) Emotiv Epoc to obtain EEG signals. To select the appropriate channels that we need in the BCI, we considered the work of Mahajan et al. [10] that use the front channels AF3 , AF4 , F3 , F4, F5, and FC6. Also, the work of Liu et al. [11] that use channels AF3 , F4, and FC6 . At the end, we chose channels AF3, F3, F4, FC5, and FC6 because we concluded that they were closely related to the human emotions.

The main feature extracted from EEG signals is the Hurst exponent. The Hurst exponent, in time series analysis, is used to identify a non-stationary behavior of EEG signals showing identifiable trends in the data. It was chosen because it showed an accuracy rate of 71.38 at work of Wang et al. [12].

Fuzzy Inference engine: It is responsible for assessing cognitive and affective aspects by applying fuzzy inferences to determine the appropriate level of the next exercise for the student. The fuzzy engine manages five fuzzy input variables: level of exercise, amount of errors, time spent in the resolution of the exercise, emotion identified through facial recognition, and valence identified by EEG recognition. The output of the fuzzy engine is the appropriate level of the next exercise. The inference engine works based on 54 fuzzy rules implemented by using library jFuzzyLogic [13]. Some of the rules are shown in Fig. *2*.

1. **IF** (level is easy) **AND** (emoEEG is positive) **AND** (mistakes is few) **THEN** (dificulty is medium)
2. **IF** (level is easy) **AND** (emoEEG is negative) **AND** (emoRostro is neutral) **AND** (mistakes is many) **THEN** (dificulty is easy)
3. **IF** (level is medium) **AND** (emoEEG is positive) **AND** (mistakes is regulars) **THEN** (dificulty is dificil)
4. **IF** (level is medium) **AND** (emoEEG is neutral) **AND** (emoRostro is happy) **AND** (mistakes is many) **THEN** (dificulty is medium)
5. **IF** (level is medium) **AND** (emoEEG is negative) **AND** (emoRostro is neutral) **AND** (mistakes is few) **THEN** (dificulty is hard)
6. **IF** (level is hard) **AND** (emoEEG is positive) **AND** (mistakes is many) **THEN** (dificulty is medium)
7. **IF** (level is hard) **AND** (emoEEG is neutral) **AND** (emoRostro is surprise) **AND** (mistakes is few) **THEN** (dificulty is hard)

Fig. 2. Fuzzy rules.

Fuzzy inference engine responds to cognitive and affective states of students. If the student commits errors but their emotional state is neutral, then we conclude that the student is focused, so the same difficulty level is maintained. If the student makes many mistakes and his emotional state is negative, a sign that he is stressed, and the level of difficulty of the exercises decreases. When the student does not make mistakes and their emotional state is neutral or positive, the level of difficulty increases.

Persistence: Is responsible for managing everything related to the database like writing, querying and updating data.

Emotional database of face expressions: It uses the affective database RaFD (Radboud Faces Database)[14].This database is a set of pictures of 67 models (including Caucasian males and females, Caucasian children, both boys and girls, and Moroccan Dutch males) displaying 8 emotional expressions (angry, disgust, fear, happy, sad, surprise, contempt and neutral). Each emotion was shown with three different gaze directions and all pictures were taken from five camera angles simultaneously. From the set, only 5 emotions were used for this work: angry, happy, sad, surprised, and neutral.

Emotional database of EEG signals: It is an affective database created from EEG signals obtained from test subjects.

3.1 ILE main interface

Figure 3 shows the interface of the ILE where the student performs the exercises presented in Java.

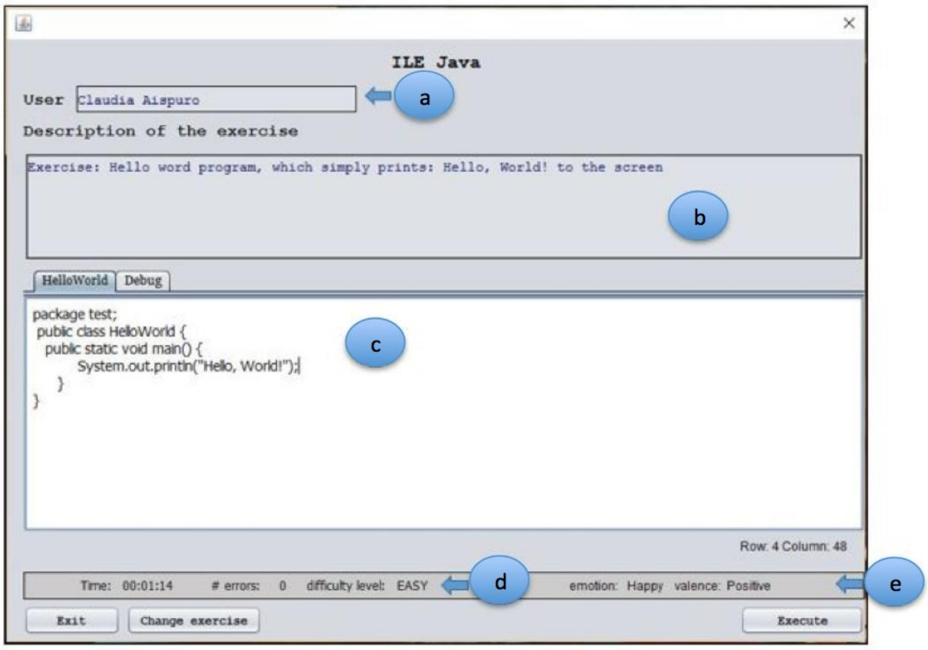


Fig. 3. ILE main interface.

The interface consists of 4 main areas, which are described below:

a. **User**. This section shows the name of the student.

b. **Description of the exercise**. This section shows the description of the exercise to be solved by the student.

c. **Code Section**. It consists of 2 tabs: one that shows the code that the student is writing to solve the exercise and another one (debug) that shows the errors produced at compile time.

d. **Cognitive aspects.** This section shows the indicators used to assess the cognitive aspect of student: time spent in the resolution of the exercise, amount of errors, and the current level of the difficulty in the exercise. There are 3 levels of difficulty to classify the exercises: Easy, Medium and Hard.

e. **Affective aspects.** This section shows the emotion and valence identified by the recognizers.

3.2 The Method

Tests were performed to ten students of the Instituto Tecnológico de Culiacán; five men and five women, who were asked not to be under the influence of caffeine or medicine at the moment of the experiment. Next, we describe each of the steps of the methodology of Bos [15] used to generate the database:

a. **Stimulus**: To evoke emotions an audiovisual medium through short videos was chosen. Recommendations found in the work of Rottenberg et al. [16] were considered to select videos.

Videos classification			
Video	Name	Emotion	Valence(Positive/Neutral/Negative)
Video 1	When Harry Met Sally	Amusement	Positive
Video 2	My Bodyguard	Anger	Negative
Video 3	Pink Flamingos	Disgust	Negative
Video 4	The Shining	Fear	Negative
Video 5	Alaska's Wild Denali	Neutral	Neutral
Video 6	The Lion King	Sadness	Negative
Video 7	Sea of Love	Surprise	Positive

shows the videos used in the experiment, its objective emotion and respective valence.

Table 1. Videos used in the experiment.

Videos classification			
Video	Name	Emotion	Valence(Positive/Neutral/Negative)
Video 1	When Harry Met Sally	Amusement	Positive
Video 2	My Bodyguard	Anger	Negative
Video 3	Pink Flamingos	Disgust	Negative
Video 4	The Shining	Fear	Negative
Video 5	Alaska's Wild Denali	Neutral	Neutral
Video 6	The Lion King	Sadness	Negative
Video 7	Sea of Love	Surprise	Positive

b. **Recording of EEG signals:** To create the emotional database each student was presented with a series of pre-selected videos with the intention to evoke a particular emotion. As the students watched the videos, the tool TestBench recorded EEG signals.

c. **Filter:** At the end of each video, students were asked to determine the video's valence (positive, negative or neutral). Then, after all videos were classified, EEG signals obtained were filtered to remove noise.

d. **Feature extraction:** We used Java library EEGFrame [17] to extract the recording features to create a feature vector.

e. **Classification:** Video classification was made by considering the expected or estimated valence (see table 1) and the valence stated by the student. The correct classification of the videos was made where those videos that did not match with labeled emotion were discarded.

Finally the affective database obtained comprises 150 records (15 for each student) classified with their respective valence.

4 Performance of the ILE Affective Java

When the system startup, the student has the option to start as a beginner (difficulty level) or take the diagnostic evaluation that will determine its initial level. The system presents a Java programming exercise according with the student's profile level. While the student is solving the exercise, the system is recording his EEG signals. When the student completes the exercise the system takes a picture of its face, and then proceeds to evaluate the exercise.

ILE displays the errors found in the program and waits for them to be corrected. When the exercise is correct, the system does the facial and EEG recognition process.

The facial recognition (See Fig. ***4Fig. 4***) consists in extracting the characteristics of the photo and then creating a feature vector, which is evaluated by a neural network trained with Weka [18] to determine the emotion with an effectiveness rate in recognition of 80%.

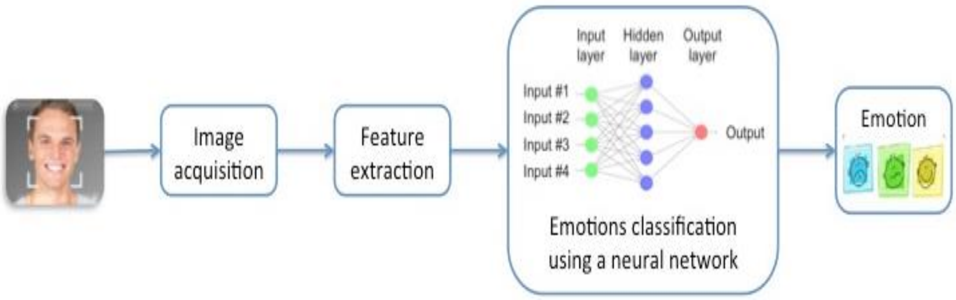


Fig. 4 . Facial recognition.

For the EEG recognition (see Fig. ***5***), the system uses Emotiv Epoc headset to register the EEG signals of the student while he is trying to resolve the exercise, then calculates the Hurst exponent, and finally performs signal classification using software from library LibSVM [19]. The library receives as an input a feature vector composed by the Hurst exponent for each of the five channels in the BCI (Emotiv) previously selected. As the result, the valence of the signal is obtained, with an effectiveness rate in recognition of 70%. We believe that if we increase the number of records of the corpus this rate can be increased.

Fig. 5. EEG recognition.

After performing the recognition, the system uses a fuzzy inference engine to define the next problem suitable for the student based on: cognitive aspects, the emotion and valence identified.

5 Conclusions and Future Work

An ILE that manages to give individual instruction to the students, based on their cognitive and affective state was implemented as a result of the integration of several components. The most important are: facial expression recognition, EEG signal recognition and fuzzy inference engine.

Future work intends to perform tests of the ILE to determine the accuracy of the emotion recognition achieved and determine if the ILE manages to help students improve their Java programming skills. It is also expected to implement other ways of emotional recognition to increase the accuracy of the recognizer.

In addition, since the tests were performed only with graduate students with previous knowledge of Java programming language, we hope to make experiments with students who do not have programming skills.

References

1. B. S. Bloom: The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. Educational Researcher, vol. 13, no. 6, pp. 4–16 (1984)
2. B. Woolf, W. Burleson, I. Arroyo, T. Dragon, D. Cooper, R. Picard: Affect-aware tutors: Recognising and responding to student affect. Int. J. Learn. Technol., vol. 4, no. 3/4, pp. 129–164 (2009)
3. R. Kanfer, P. L. Ackerman: Motivation and cognitive abilities: An integrative/aptitude-treatment interaction approach to skill acquisition. Journal of ,Applied Psychology, vol. 74, no. 4, pp. 657–690 (1989)
4. R. S. J. D. Baker, S. K. D'Mello, M. M. T. Rodrigo, A. C. Graesser: Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based

learning environments. Int. J. Hum. Comput. Stud., vol. 68, no. 4, pp. 223–241 (2010)
5. T. Jenkins: On the Difficulty of Learning to Program. Language (Baltim), vol. 4, pp. 53–58 (2002)
6. S. K. D'Mello, A. Graesser: Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. User Model. User-Adapted Interact., vol. 20, no. 2, pp. 147–187 (2010)
7. I. Roll, V. Aleven, B. M. McLaren, K. R. Koedinger: Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. Learn. Instr., vol. 21, no. 2, pp. 267–280 (2011)
8. S. D. Mello, A. Olney, C. Williams, P. Hays: Gaze tutor : A gaze-reactive intelligent tutoring system. J. Hum. Comput. Stud., vol. 70, no. 5, pp. 377–398 (2012)
9. R. Z. Cabada, M. Lucía B. Estrada: Ambiente inteligente de aprendizaje con manejo afectivo para Java. Research in Computing Science, vol. 92, pp. 111–121 (2015)
10. R. Mahajan, D. Bansal, S. Singh: A Real Time Set Up for Retrieval of Emotional States from Human Neural Responses. International Journal of Medical, Health, Pharmaceutical and Biomedical Engineering, vol. 8, no. 3, pp. 142–147 (2014)
11. Y. Liu, O. Sourina, M. K. Nguyen: Real-time EEG-based Emotion Recognition and its Applications. Trans. Comput. Sci. XII, vol. 6670, pp. 256–277 (2011)
12. X.-W. Wang, D. Nie, B.-L. Lu: Emotional state classification from EEG data using machine learning approach. Neurocomputing, vol. 129, pp. 94–106 (2014)
13. P. Cingolani, J. Alcalá-Fdez: JFuzzyLogic: A robust and flexible Fuzzy-Logic inference system language implementation. In: IEEE International Conference on Fuzzy Systems (2012)
14. O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, A. van Knippenberg: Presentation and validation of the Radboud Faces Database. Cogn. Emot., vol. 24, no. 8, pp. 1377–1388 (2010)
15. D. O. Bos: EEG-based Emotion Recognition - The Influence of Visual and Auditory Stimuli. Emotion, vol. 57, no. 7, pp. 1798–806 (2006)
16. J. Rottenberg, R. D. Ray, J. J. Gross: Emotion elicitation using films. Handbook of emotion elicitation and assessment. In: Handbook of emotion elicitation and assessment Series in affective science, pp. 2–28 (2007)
17. A. Jovic, L. Suc, N. Bogunovic: Feature extraction from electroencephalographic records using EEGFrame framework. In: Proceedings of the 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2013), no. 6596396, pp. 965–970 (2013)
18. I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, S. J. Cunningham: Weka : Practical Machine Learning Tools and Techniques with Java Implementations. Seminar, vol. 99, pp. 192–196 (1999)
19. C.-C. Chang, C.-J. Lin: LIBSVM: A Library for Support Vector Machines. ACM Trans. Intell. Syst. Technol., vol. 2, pp. 27:1–27:27 (2011)

Java Tutoring System with Facial and Text Emotion Recognition

Ramón Zatarain-Cabada, María Lucia Barrón-Estrada, Jorge García-Lizárraga, Gilberto Muñoz-Sandoval, José Mario Ríos-Félix

Instituto Tecnológico de Culiacán, Culiacán Sinaloa, Mexico

{rzatarain, lbarron, m86170331, m14170101, mario_rios}@itculiacan.edu.mx

Abstract. This paper presents the design and implementation of an intelligent tutoring system (ITS) for teaching JAVA, which can recognize the user's emotional state through facial expressions and textual dialogues. For facial emotion recognition we implemented a neural network with WEKA library and a facial feature extractor with OPENCV library. The ITS applies a semantic algorithm (ASEM) to extract textual emotions through dialogues, which has shown a degree of assertiveness of 80% in tests for graduate students. In addition, the tutor uses a set of fuzzy rules to determine the complexity of the next exercise, considering the program implementation time, program executions and compilations, and current difficulty level.

Keywords: Affective computing, intelligent tutoring system, neural networks, fuzzy systems.

1 Introduction

In recent decades, technology has had a huge development by allowing the design of more complex programs, by using new software development techniques, new data mining algorithms, and more efficient software architectures. Likewise, the integration of the emotional state of the user with technologies like neural networks, genetic algorithms, and fuzzy logic produces in the case of educational software a new way of permeating awareness among students and software.

At the beginning Intelligent Tutoring Systems used to be implemented with traditional principles of behaviorism, moving later to other more interactive and dynamic learning theories, in which the subjects interact in virtual learning environments. Affective Tutoring Systems (ATS) are intelligent systems that incorporate the ability to recognize the emotional state of the students allowing the user to interact with exercises that stimulate their emotional state [1].

Given the growing demand for learning programming languages, it is necessary to use new tools using modern techniques such as emotion recognition, which motivate and facilitate student learning. At present, supporting tools for teaching programming languages only take into account cognitive aspects of the student, ignoring other behavior features like user´s emotions. Studies have shown that emotions are closely related to cognitive processes such as learning [2].

This work integrates diverse technologies into the affective tutor named “Java Zenzei”, such as the use of a neural network (using WEKA) and a feature extractor for recognizing facial emotions, as well as a semantic algorithm (ASEM) to extract emotions from textual dialogue.

This paper is divided as follows. The second section describes related work. The third section mentions the effort done in the development of a bimodal recognizer that was used in our research. The fourth section describes the tutoring system operation. The fifth section shows results of the proposed work. Finally, the sixth section presents conclusion of the work.

2 Related Work

There exists a lot of research about multimodal recognition to detect emotions taking into account different aspects. For example, Soleymani [3] combines EEG signals with eye tracking that can be considered bimodal recognition. There are also a lot of work with multimodal recognition like MERT [4] that works with face, voice, and body motion.

In another paper, Sebe, Cohen, and Huang [5] use web camera and microphone to improve the human-computer interaction, where they state that facial gestures and voice are crucial for that interaction resulting in different benefits for their research. D'Mello [6] has also contributed in the field of multimodal recognition with his project AutoTutor [7] which uses voice, body motion, and facial gestures to identify emotions.

A recent version of AutoTutor named Affective AutoTutor [10] adds capabilities to recognize the affective state of a student by using facial expressions, body language (posture) and textual dialogue recognition. It detects when the student is bored, confused or frustrated and intervenes through a pedagogical agent.

Binaly [11] describes four different text based emotion recognition techniques, Keyword spotting method, Lexical Affinity Method, Learning based method and hybrid methods.

The main contribution of this work is the use of a bimodal emotion recognition (facial and Spanish textual recognition) inside of an intelligent tutoring system for learning Java.

3 Bimodal Recognition of Emotion

The emotion recognition is a complex task and to date the rates of most recognizers do not get 100% of success. So using bimodal recognition is a way to increase the rate of success in the recognition. Next we explain the structure of the Java Zenzei Tutor.

3.1 Java Zenzei Structure

The system has a Log in component which identifies and authenticates the user. There’s a component to get the user´s current difficulty level by searching at the database, in case the user is new, a diagnostic test module is executed, which shows a questionnaire to be solved, to determine the starting level of the user.

The system shows an exercise based on the user´s current level. When the exercise is solved, the exercise´s variables (exercise validation, amount of compilations and time required) are obtained, also the emotional state variables (facial and text emotion) are extracted by taking a photo of the user and asking a question about the exercise.

Next the system uses a fuzzy logic engine to obtain a new user´s level by combining the current level, exercise´s variables, and the emotional state variables. The process to obtain a new level is performed every time the user completes an exercise.

The ATS structure (figure 1) presents a clear idea of how the system works.

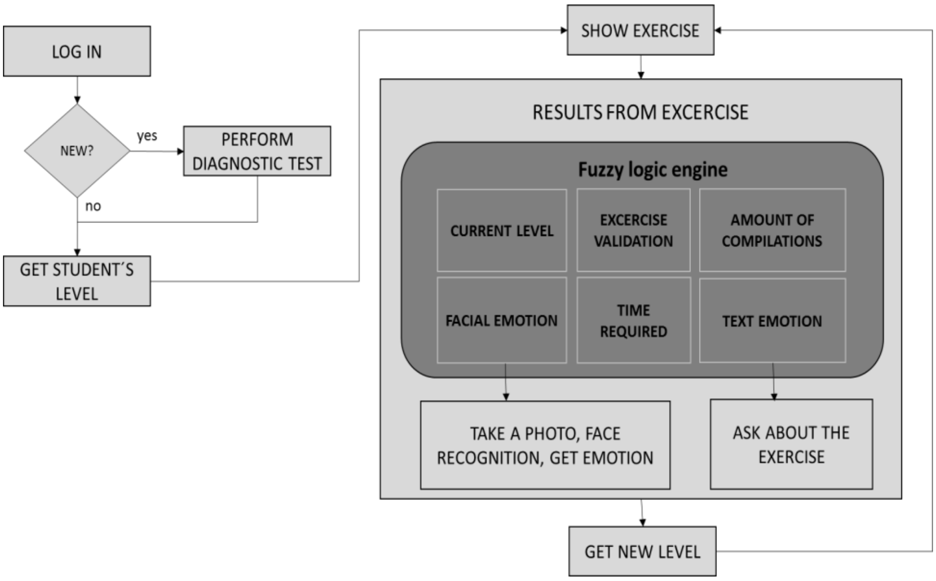


Fig. 1. ATS Structure.

Persistence of the information required by the tutor like student´s information, exercises, answers for each question, information of each exercise to be executed, is stored in an object-oriented database (DB4O).

The recognition method for emotions works with two recognizers, where the first one is a method for emotion recognizing through face detection. In this process, the system take a picture of the student when he/she is solving the exercise. We use a feature extractor for the face which was implemented with OPENCV library and a neural network implemented in WEKA for classifying emotions. The emotions identified by the facial recognizer are: joy, surprise, sadness, anger and neutral emotion.

The system uses a set of fuzzy rules to identify the complexity of the next exercise where six aspects are considered: student´s level (beginner, basic, intermediate, and advanced), validation of last exercise, number of compilations, identified facial emotion, time spent in exercise and identified text emotion. We used a Java library named JFuzzyLogic for implementing the fuzzy system and a new semantic algorithm (ASEM) for textual dialogue. The evaluation allows the ATS to identify the complexity level of the next exercise which can be lesser, greater or equal to the current level. An example of a fuzzy logic rule is shown next (figure 2).

```
IF currentLevel IS intermediate  AND
   elapsed Time IS few           AND
   compilations IS regular       AND
   validation IS correct         AND
   facialEmocion IS joy          AND
   textualEmotion IS joy         THEN  level IS advanced;
```

Fig. 2. Fuzzy logic rule example.

The proposed emotion recognition algorithm through the text is described below.

3.2 ASEM Algorithm

For emotion recognition from text, we implemented a semantic algorithm (ASEM) that allows identifying emotions from text dialogues using semantic labels [8], where the user writes different comments for answering questions received in a random form. This semantic algorithm is based mainly in a word corpus called SEL [9] which was built by experts with an output emotion and a probability factor of affection (PFA).

The ASEM algorithm incorporates semantics rules that allow emotion detection with a rate of success greater than 80% according to different test performed with graduate students. The rate of success increases by adding new words together with the corresponding PFA which must be computed by an expert. The emotion recognition is made in two phases: training and application.

During the training phase, we used an interaction which allows the communication with student, where some processing text is recorded. Next the emotion is generated according to the existing words found in the original corpus, showing the parameters used in the emotion generation, and also those words which could not be found in the corpus (they are also added to the corpus NewWords). For each new added word an expert define the PFA. This first phase is realized in an iterative form until the new corpus has found the expected success level.

In the second phase (Application) ASEM perform the next steps (figure 3):

- A student input a text line (input dialogue).
- The text is normalized: the accented words, numbers, and special characters are removed and uppercase letters are converted to lowercase.
- Non emotion words like *he*, *she*, *the*, etc. are removed by using the corpus *StopWords*.
- Semantic words are sought in corpuses *Semantic* and *improper words*. If semantic words are not found in the corpuses the new words are added to corpus *NewWords*.

- If the semantic word is found in the corpuses, the word features (PFA and emotion) are extracted.
- The emotion in classified according to the features of the word.
- The emotion with greatest intensity is produced.

Figure 3 shows the ASEM algorithm that can recognize the emotion from text.

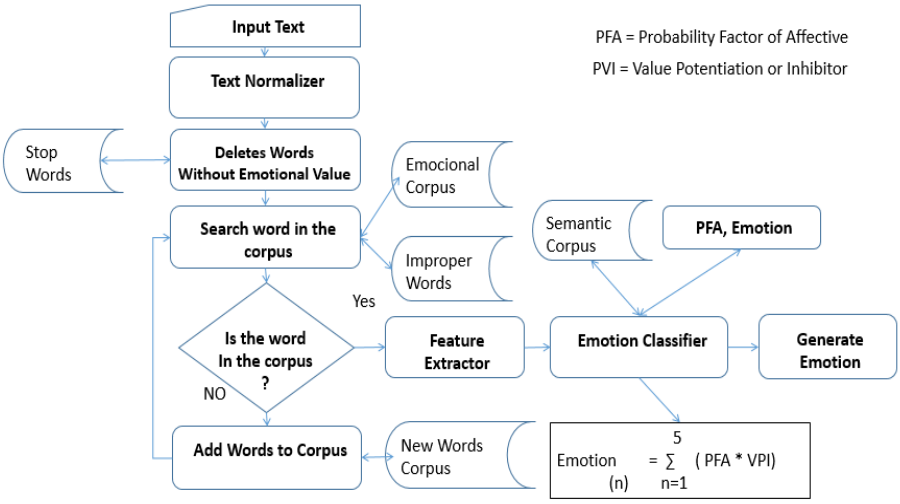
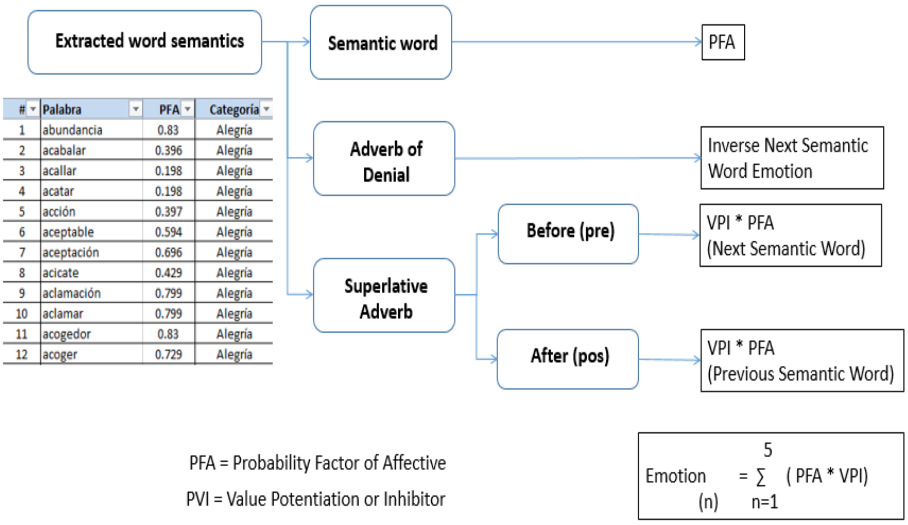


Fig. 3. ASEM algorithm.



#	Palabra	PFA	Categoría
1	abundancia	0.83	Alegría
2	acabalar	0.396	Alegría
3	acallar	0.198	Alegría
4	acatar	0.198	Alegría
5	acción	0.397	Alegría
6	aceptable	0.594	Alegría
7	aceptación	0.696	Alegría
8	acicate	0.429	Alegría
9	aclamación	0.799	Alegría
10	aclamar	0.799	Alegría
11	acogedor	0.83	Alegría
12	acoger	0.729	Alegría

Fig. 4. Emotion detection.

To get the emotion we use PFA and EIV (enhancer/inhibitor) values for every word found in the text written by the students. The whole process is explained next (figure 4):

- The next semantic word is obtained.
- This word is sought in corpus Semantic. If found, the emotion and PFA value are extracted.
- If the word is a negative adverb, the emotion of the next word is switched (e.g. happy to sad).
- If the word is a superlative adverb, we identify the word as a pre-adverb or a post-adverb. In case of being a pre-adverb we enhance the PFA, and EIV of the next semantic word. In case of being a post-adverb we enhance the PFA and EIV of the previous last semantic word.
- Finally we use the sum of each found emotion in the text dialogue, and then we determine the emotion by the greatest value.

Next, we give some details produced in the emotion recognizing from text:

- There are many words that were not found in the original corpus (2,036 words) from a total of more than 300,000 of the Spanish language.
- Most students don't use accent in words.
- There are words with no emotion (neutral emotion). These words integrate a corpus named StopWords (e.g. *the, he, or she*)
- Improper words are eliminated from the text but they are considered as having a negative emotion.
- The emotions identified by the text recognizer are *joy, surprise, neutral, sadness and anger.*
- Words that deny a sentence (e.g. *no* or *never*) change its meaning (valence) and they are already stored in the semantic corpus.
- Words which define the intensity of the words (e.g. *very, few, nothing*) are already stored in the semantic corpus.

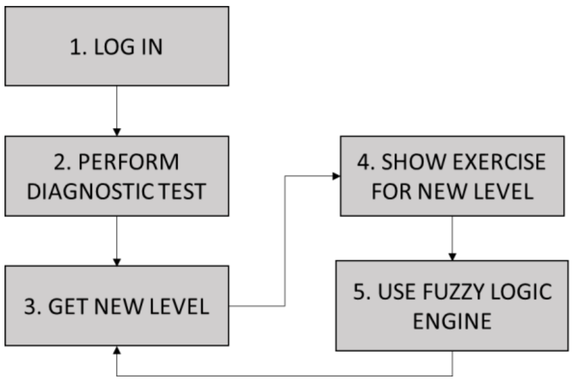


Fig. 5. Basic operation of the tutor.

4 A Working Session with the ATS Java Zenzei

The ATS starts with a step of login for authentication and registration of users. In case of a new user the system proceeds to realize a diagnostic test, which consist in a set of multiple choice questions. The result of the diagnostic test is the initial difficulty level of the student. The next step presents an exercise for programing in the Java language. As we explain before in section 1, a set of variables related to the solving of the problem define the new level of the student and the new exercise (see figure 5).

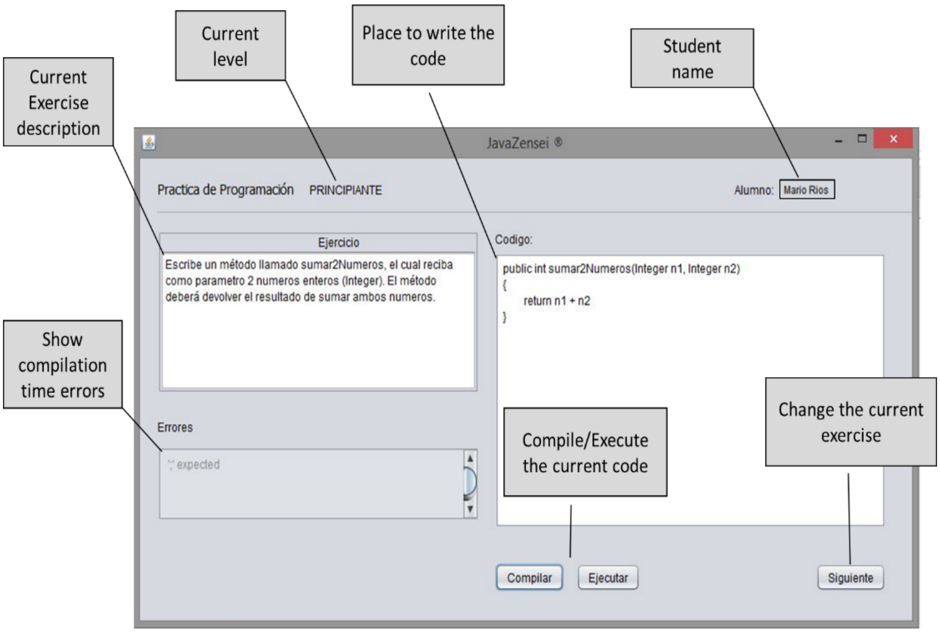


Fig. 6. Main screen of tutor JavaZenzei.

Once the student solve the exercise, a fuzzy inference engine defines the next exercise using a set of variables previously mentioned. The main interface of the Java Zenzei is shown in figure 6 with some brief explanation given in gray boxes.

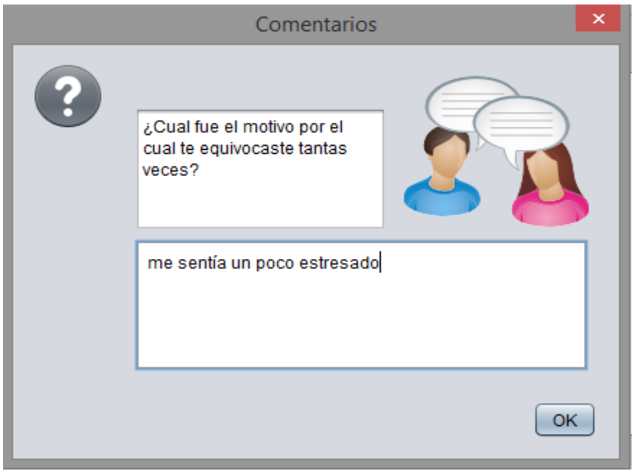


Fig. 7. Window to write comments on the exercises.

When a student ends an exercise a small windows is displayed asking to add comments related to that exercise (see figure 7).

5 Experiments and Results

We perform some initial test with graduate students from the Instituto Tecnológico de Culiacan. The exercises were designed for students with basic knowledge of Java language. In the test there were 9 students who finished the exercises in a short time and with few errors. That allowed students to move faster to more complex exercises. Table 1 shows the results obtained from the test.

Table 1. Result of the test.

Student	Gender	Diagnostic Level	Average Time	Failed Compilations	Average Emotion	Reached Level
E01	M	Basic	2.3 min	2	Neutral	Intermediate
E02	M	Basic	3.5 min	4	Surprise	Basic
E03	F	Basic	2.6 min	3	Neutral	Intermediate
E04	M	Advanced	2.1 min	0	Happy	Advanced
E05	M	Intermediate	2.3 min	2	Neutral	Advanced
E06	M	Intermediate	2.2 min	1	Happy	Advanced
E07	F	Basic	2.5 min	1	Neutral	Intermediate
E08	M	Intermediate	2.4 min	3	Surprise	Intermediate
E09	M	Intermediate	2.3 min	1	Neutral	Advanced

At table 1 we can see the student E06 for example, started at an intermediate level, had a short time solving the exercises, got just 1 compilation error, and the average emotion found was happy, so eventually the ITS promoted him to advanced level.

The rate of success of the emotion recognition was 80%. Whenever the student comments had more than 3 words that couldn't be found on the corpus SEL the emotion recognizer produce an incorrect results (incorrect emotion). The emotion recognizer achieves better results by adding those words in the corpus SEL. Finally whenever there was no match between the facial and textual emotion we choose the textual emotion as the correct one because most of the time the text emotion recognizer produce better results.

6 Conclusions

In this paper we presented a novel intelligent tutoring system for the support of learning the Java language. This system uses two emotion recognizers that work through text and facial expressions. We made different tests of the ATS with graduated students of computer science, producing the next results: the tutoring system based on facial expression is able to recognize emotions with a success rate of

80%. The emotion recognizer based on text dialogues achieved a success rate of 85%. The fuzzy system always produced the adequate exercises according to stablished parameters. For example, if the student was boring, had few errors, had few compilation and executions errors, and had a short time to solve the problem, the system increase the complexity of the exercises. On the other hand, if the student was not boring, had few errors and a greater time of solving the problem, the complexity of the next exercises remains the same. But if the student had too many errors and spent a lot of time solving the exercise, then the system decrease the complexity level of the next exercise.

Based on the obtained results we stablish that combining both emotion recognizers (facial expression and text dialogues), add precision to the work of identifying the emotional state of a student. We found that the precision of the text-dialogue recognizer mostly depends on the number of words stored in the corpus Semantic.

As future work we expect to add more words to corpus Semantic, so we can increase the rate of success of the recognizer. Another upcoming work is to integrate more exercises with different rates of complexity with the goal of having a more complete ATS.

References

1. Hernández, Y., Sucar, L. E., Arroyo-Figueroa, G.: Building an affective model for intelligent tutoring systems with base on teachers expertise. In: Proc. of MICAI, Springer, pp. 754–764 (2008)
2. Belavkin, R.: The role of emotion problem solving. In: C. Johnson (Ed.), Proceedings of the AISB ’01 Symposium on emotion, cognition, and affective computing, Heslington, York, England, pp. 49–57, (2001)
3. Soleymani, M.: Multimodal Emotion Recognition in Response to Videos. IEEE transactions on affective computing, vol. 3, no. 2 (2012)
4. Bänziger, T., et al.: Emotion Recognition from Expressions in Face, Voice, and Body: The Multimodal Emotion Recognition Test (MERT). American Psychological Association (2009)
5. Sebe, N., Cohen, I., Huang. T.: Multimodal Emotion Recognition. WSPC (2014)
6. D’Mello, S., Graesser A.: Multimodal Semi-Automated Affect Detection from Conversational Cues, Gross Body Language and Facial Features. Springer Science+Business Media B.V (2014)
7. Graesser, A.: AutoTutor: an intelligent tutoring system with mixed initiative dialogue. IEEE Transactions on Education, Vol. 48 (2005)
8. Wu, C. H., Chuang, Z. J., Lin, Y. C.: Emotion recognition from text using semantic labels and separable mixture models. ACM transactions on Asian language information processing (TALIP), 5(2), 165–183 (2006)
9. Chaffar, S., Inkpen, D.: Using a heterogeneous dataset for emotion analysis in text. In Advances in Artificial Intelligence, pp. 62–67, Springer (2011)
10. D’Mello, S., Graesser, A.: AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 2, no. 4 (2012)

11. Binali, H., Wu, C., Potdar, V.: Computational Approaches for Emotion Detection in Text. In: IEEE DEST (2010)

A Framework for Automatic Identification of Learning Styles in Learning Management Systems

Ignacio Núñez Márquez, Luis-Felipe Rodríguez, Guillermo Salazar Lugo, Luis A. Castro, Manuel Domitsu Kono

Instituto Tecnológico de Sonora, Department of Computer and Design, Mexico

ignacio_nmarquez@hotmail.com, {luis.rodriguez, gsalazar47040, luis.caztro, manuel.domitsu}@itson.edu.mx

Abstract. A current concern of modern societies is the quality of their educational systems. Learning Management Systems (LMSs) are technological tools that have been used to improve and extend the traditional educational system. Although LMSs have greatly benefited from technological advances, this type of educational platforms are still in need of mechanisms to provide virtual educational environments in which students are considered the main actor in the design of the learning process, thus contributing to increase the quality of educational systems. In particular, LMSs usually lack mechanisms for recognizing users' learning styles, which describe the way a learner acquires and process information. In this work we propose a framework for automatic identification of learning styles in LMSs and present a specific implementation. A key goal of the framework design is to provide researchers and practitioners with a tool that facilitates the specification of expert knowledge for classifying students with respect to their learning styles in LMSs.

Keywords: Learning style, LMS, automatic identification, framework.

1 Introduction

Education is a key aspect associated to the development of a country. The knowledge acquired by individuals in educational institutions impact on a country's capability for doing research, innovation and technological development as well as on its economic growth [14]. Educational systems are therefore designed to facilitate individuals with access to education and to allow them to acquire abilities and knowledge that can contribute to their own personal and academic development and to the progress of their countries [2]. Moreover, a crucial concern of modern societies is the quality of educational systems. This desire for quality in education becomes a challenge as educational systems are mainly based on traditional models that consider the instructor as the main actor, minimizing the role of students as individuals with particular needs and ways of learning.

This situation has motivated researchers to find novel strategies and methods aimed at promoting the quality of educational systems [15]. For example, the availability of mobile devices that can access the Internet and the institutions' technological infrastructure has open avenues for novel designs of educational methods based on online learning, social networking tools, and intelligent classrooms [1, 13]. In particular, LMSs are tools that have greatly benefited from technological advances and have become a mechanism for improving and extending traditional educational systems [1, 16, 17]. LMS are technological tools that enable the creation of virtual learning environments that include components for adding and displaying academic material (e.g., lectures, exercises and tests), communication, defining sequences of activities, among other things. LMSs also include mechanisms to monitor learners' academic behavior and to make these data available to teachers in order to mediate the learning process.

Although LMSs have taken advantage of technology to manage various types of content (e.g., multimedia), to offer a diversity of communication mechanisms, complex graphics and efficient user-behavior tracking mechanisms, LMSs are still in need of mechanisms to provide virtual educational environments in which learners are considered the main actors in the design of their learning process, thus contributing to increase the quality of educational systems [8, 16]. This concern becomes crucial in the academic development of the individual as it has been recognized that learners have differences in how they acquire and process information [5,10]. There are several ways for involving learners in the design of their learning process. For example, monitoring learners' academic performance and emotions may inform how learners perceive their learning process and help to infer their academic interests. Ultimately, this information becomes useful to personalize the user's learning environment in a LMS. Although several efforts have been reported in the literature, these proposals are usually designed and developed for very specific academic purposes and validated in very controlled environments [7,11]. The redesign and implementation of these kinds of proposals in LMSs such as Moodle and Chamilo would require considerable effort.

In this paper, we propose a framework for automatic identification of learning styles in LMSs such as Moodle. A key goal of the framework design is providing researchers and practitioners in the field of education with a tool that facilitates the specification of expert knowledge for classifying students with respect to their learning styles in educational platforms. We also present an implementation of the framework, which takes advantage of theories and models of learning styles reported in psychology and education literature, techniques from the field of Artificial Intelligence (AI), and advances in technologies for distributed systems. The proposed framework is not intended to define a mechanism to automatically identify users' learning styles with a minimal classification error. Instead, the presented framework attempts to serve as a guideline for the generation of virtual learning environments that take advantage of existing technologies and related literature for the automatic identification of users' learning styles in LMSs.

2 Related Work

User-centered learning environments aim at minimizing the weaknesses of traditional educational systems that follow the "One-Size-Fits-All" model [9]. Personalized learning environments are designed to involve learners in the design of their own learning process and allow them to set their own goals for learning. In these types of environments, learners progress at their own pace and are supported to reach their maximum potential by providing access to a wide range of academic material and teaching strategies, according to learners' strengths and weaknesses, and academic and personal interests [4]. A personalized learning environment consists of two phases:

1. Understand learners' "*situation*" in terms of aspects such as their cognitive and affective state, previous knowledge, abilities, personal and academic interests, and learning style.
2. Once learners' "*situation*" is known, it is necessary to personalize students' learning environment.

It has been recognized that individuals have preferences regarding the type of content they use for learning [5,10]. For example, some students prefer watching videos rather than reading lectures. A key indicator that describes users' preferences is their learning style. In general, the learning style is the way individuals learn [7]. Learning styles allow classifying learners' behavior according to how they take the information, how they form strategies to learn, how they understand new concepts, and how they analyze information used to learn a particular knowledge. The literature reports several studies about learning styles in which learners are usually classified according to a series of categories. For example:

- Felder and Silverman [5] explain students' learning preferences based on four dimensions: active and reflexive learners; sensing and intuitive learners; visual and verbal learners; and sequential and global learners;
- Kolb [10] propose a model to explain learning styles that is based on four categories: Diverging, Assimilating, Converging, and Accommodating.

LMSs like Moodle aim at supporting teachers in creating and managing courses and provide them with a great variety of features that can be included in a course, such as learning material, quizzes, discussion forums, and assignments. Moreover, these types of LMSs provide a set of features to support teachers in the construction, administration and management of courses. Most LMSs treat all learners equally, regardless of their learning style preferences [8,16]. Recognizing users' learning styles may bring many benefits in LMSs. In LMSs like Moodle and Chamilo, understanding how a student learns makes it possible to personalize the virtual learning environment by determining which elements comprise such learning environment, including teaching strategies, academic material, learning activities, feedback strategies, and communication strategies. Although most LMSs provide educators, administrators and learners mechanisms for personalizing the learning environment, this personalizing process must be carried out by the instructor, meaning it does not occur automatically.

The literature reports various attempts to create computer systems that help with the automatic identification of users' learning styles [7, 8]. The traditional way is based on the use of questionnaires that learners fill out. However, this strategy has been criticized as questionnaires usually include more than 100 items, making it tedious for learners to answer the questions or inducing learners to answer them arbitrarily. The automatic identification approach consists on monitoring learners' behavior and building a user model that describes their preferences. A model for learner classification is then built based on the user model and the results of applying a learning style instrument to a group of initial learners. This model is then used to automatically classify learners on the basis of their behavior and without the need to answer the questionnaire [7].

Although current technology has enabled the construction of highly complex educational platforms that may help users learn different things, mechanisms for automatically identifying users' learning styles in these platforms are still to be developed. Moreover, most efforts for automatic identification of learning styles are developed as prototypes that are validated in very controlled environments [7], making them unsuitable for their use in LMSs like Moodle.

3 Proposed Framework

In this section we present a framework that endows LMSs with mechanisms to automatically identify users' learning style. The framework is designed to take advantage of complex characteristics offered by existing LMSs such as Moodle, which are tools that have greatly benefited from technological advances.

The idea behind this proposal is to provide a computer-based service that classifies users of LMSs with respect to their learning style. A key assumption in the design of this computer-based service is that the behavior of LMS' users can be analyzed on the basis of expert knowledge drawn from theories of learning styles reported in psychology and education literature. In particular, this computer-based service is designed to identify users' learning styles taking into account the following two types of information:

1. User behavior in a LMS: The framework is designed to take into account the information a LMS is able to monitor about users' behavior and that is useful for classifying their learning styles, such as courses completed, activities initiated and completed, resources visited, test results, the use of forums and chats, pages accessed and the time and date learners accessed them, and the type of content revised by learners such as text, images or videos.
2. Expert knowledge about learning style models: The framework uses expert knowledge in the classification process. It is designed to take advantage of existing theories and models that explain how learners can be classified in a variety of categories according to how they take in and process information. This knowledge may be represented by experts in a variety of ways (e.g., in the form of simple IF-THEN rules).

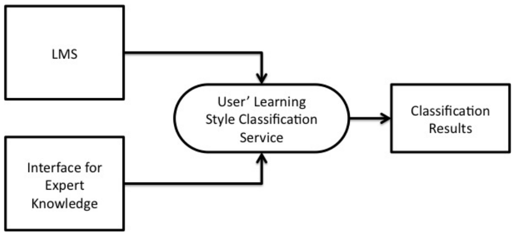


Fig. 1. Framework for the automatic identification of users' learning styles in LMSs.

Figure 1 presents the main components included in the framework and their interrelation: LMS, Interface for Expert Knowledge, User's Learning Style Classification Service, and Classification Results. The data flow between these components is as follows. The *User's Learning Style Classification Service* component is responsible for the classification of users' according to their learning style. This component receives data from the *LMS* and *Interface for Expert Knowledge* components. The *LMS* component sends information about users' behavior within an educational platform. This information is then analyzed on the basis of information sent from the *Interface for Expert Knowledge*. This last component allows human experts to indicate how the behavior of a learner (within the context of a particular LMS) should be analyzed in order to infer the student's learning style. It is important to note that the framework design recognizes that LMS differ in the type of information they are able to monitor and that the representation of expert knowledge may be based on a variety of learning style models. Finally, the *User's Learning Style Classification Service* component sends the classification results to the *Classification Results* component, which presents information about the classification process and the preferences of learners based on their specific learning style. The results presented by these components can be then used to personalize the virtual learning environment in LMSs.

4 An Implementation of the Proposed Framework

As previously mentioned, one of the main objectives of the proposed framework is to take advantage of existing LMSs. We believe that these are tools that have greatly benefited from technological developments and that have proven useful in the education domain. In this sense, the *LMS* component included in the framework can be replaced by any LMS. For example, Moodle provides mechanisms to generate reports of learners' activities, such as resources visited, test results, and the use of forums and chats. All these data becomes useful for identifying user preferences and interests. Moreover, it is possible to add additional features and functionality to Moodle via plugins, which can be developed to automatically monitor and send data to the classification component.

According to the proposed framework, models of learning styles reported in the literature are useful to define how to analyze the information monitored in a LMS like Moodle. In Section 2, we mentioned the models proposed by Felder and Silverman [5] and Kolb [10]. In general, all models classify students according to a series of categories that explain how learners acquire and process information. These models include a test or questionnaire to identify students' learning styles. More importantly, these models provide explanations of learners' academic behavior and preferences. As shown in Table 1, descriptions provided by models can be represented in terms of IF-THEN rules. A rule can be defined for a visual learner: IF the learner usually visit resources of type pictures AND the learner usually describe activities using diagrams THEN the learner tends to be VISUAL. Similarly, IF-THEN rules can be defined to determine users' learning styles based on their behavior. The *Interface for Expert Knowledge* can then be designed to introduce expert knowledge in terms of rules. Moreover, an implementation of the *User's Learning Style Classification Service* component may take advantage of these rules for classifying the data sent from the *LMS*.

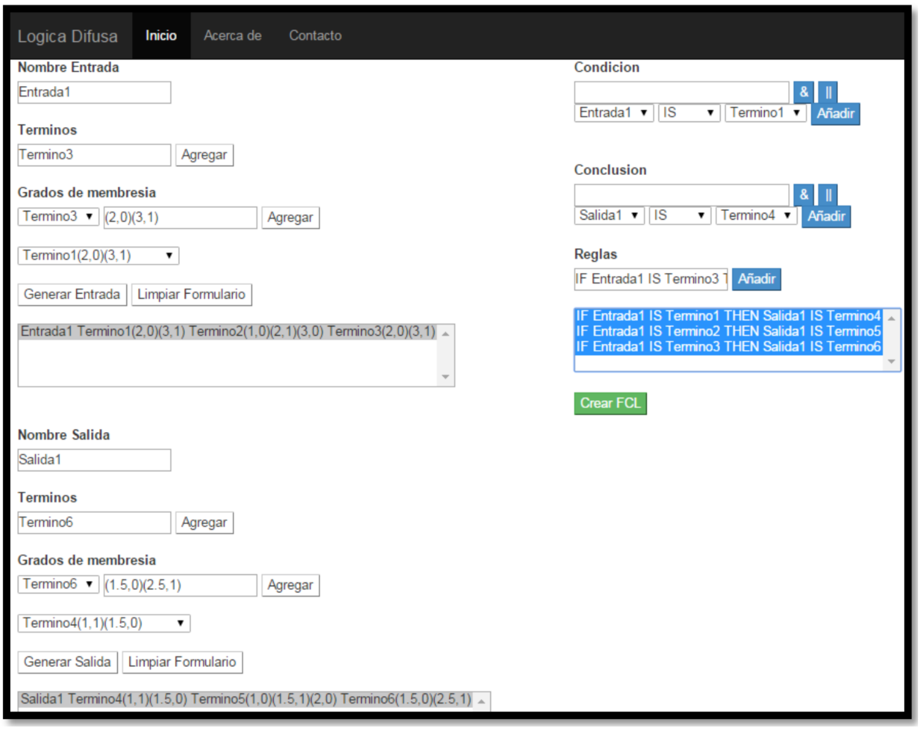


Fig. 2. Interface implemented to introduce Expert Knowledge.

We used Fuzzy Classification to implement the *User's Learning Style Classification Service* component. Fuzzy Classification is based on the theory of fuzzy sets and fuzzy logic. This type of classification assigns learners into a fuzzy

```
FUNCTION_BLOCK FCL

VAR_INPUT
    Entrada1 : REAL;
END_VAR

VAR_OUTPUT
    Salida1 : REAL;
END_VAR

FUZZIFY Entrada1
    TERM Termino1 := (2,0)(3,1);
    TERM Termino2 := (1,0)(2,1)(3,0);
    TERM Termino3 := (2,0)(3,1);
END_FUZZIFY


DEFUZZIFY Salida1
    TERM Termino4 := (1,1)(1.5,0);
    TERM Termino5 := (1,0)(1.5,1)(2,0);
    TERM Termino6 := (1.5,0)(2.5,1);
END_DEFUZZIFY


RULEBLOCK No1
    AND : MIN;
    ACT : MIN;
    ACCU : MAX;
    Rule 1:   IF Entrada1 IS Termino1 THEN Salida1 IS Termino4;
    Rule 2:   IF Entrada1 IS Termino2 THEN Salida1 IS Termino5;
    Rule 3:   IF Entrada1 IS Termino3 THEN Salida1 IS Termino6;
END_RULEBLOCK

END_FUNCTION_BLOCK
```

Fig. 3. An example of a FCL file.

set. Importantly, rule-based fuzzy classifiers may be constructed by specifying classification rules as those described in the previous paragraph. A more detailed description of Fuzzy classification is out of the scope of this paper (see [12]).

We used the jFuzzyLogic library [3] to implement fuzzy classification in the *User's Learning Style Classification Service* component. This library is written in Java and implements the Fuzzy Control Language (FCL). In jFuzzyLogic, all necessary information for the classification task is included in a FCL file: linguistic variables (used to represent variables that describe user behaviors and output variables that represent learning style categories), linguistic terms,

Table 1. Learning style categories and their descriptions (Felder and Soloman [6]).

Active and reflexive learners	Active learners tend to retain and understand information best by doing something active with it–discussing or applying it or explaining it to others. Reflective learners prefer to think about it quietly first
Sensing and intuitive learners	Sensing learners tend to like learning facts and often like solving problems by well-established methods and dislike complications and surprises. Intuitive learners often prefer discovering possibilities and relationships, they like innovation and dislike repetition.
Visual and verbal learners	Visual learners remember best what they see (pictures, diagrams, flow charts, time lines, films, and demonstrations). Verbal learners get more out of words (written and spoken explanations).
Sequential and global learners	Sequential learners tend to gain understanding in linear steps, with each step following logically from the previous one. Global learners tend to learn in large jumps, absorbing material almost randomly without seeing connections, and then suddenly "getting it"

membership functions (to express in which degree a leaner belongs to a given Fuzzy subset), linguistic rules (the IF-THEN rules), and other parameters used for the classification task (e.g., the defuzzifier method).

Figure 2 and Figure 3 show the interface implemented to introduce Expert Knowledge and the FCL that results from these data. Once the data sent from the LMS is evaluated by the *User's Learning Style Classification Service* using this data included in the FCL, the classification data is sent to the *Classification Results* component to presents the results. Figure 4 shows all the components implemented and their interrelations.

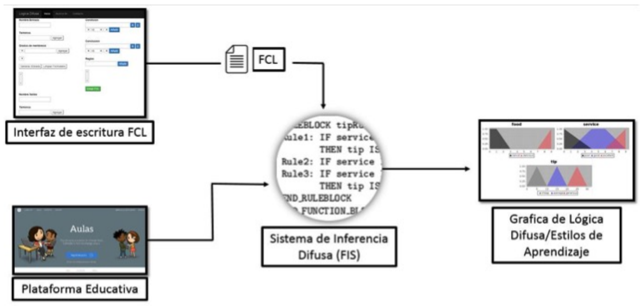


Fig. 4. A specific implementation for the proposed framework.

5 Conclusions

In this paper we presented a framework for automatic identification of learning styles in LMSs. We presented a specific implementation for the framework that takes advantage of existing LMSs, models of learning styles reported in psychology literature, and AI tools like fuzzy classification. The proposed implementation shows the validity of the framework from a technological perspective. As future work, we are planning to extend the framework implementation by including a learning mechanism in the classification phase as well as by including a second phase that deals with the automatic personalization of LMSs. Furthermore, the proposed framework must be validated by learners and teachers.

Acknowledgments. This work was partially funded by the SEP-PRODEP through the project ITSON-PTC-089.

References

1. Al-Zoube, M.: E-learning on the cloud. Int. Arab J. e-Technol. 1(2), 58–64 (2009)
2. Arnove, R.F.: Comparative education and world-systems analysis. Comparative Education Review pp. 48–62 (1980)
3. Cingolani, P., Alcala-Fdez, J.: jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation. In: FUZZ-IEEE. pp. 1–8. Citeseer (2012)
4. Dabbagh, N., Kitsantas, A.: Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. The Internet and higher education 15(1), 3–8 (2012)
5. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. Engineering education 78(7), 674–681 (1988)
6. Felder, R.M., Soloman, B.A.: Learning styles and strategies. At URL: http://www. engr. ncsu. edu/learningstyles/ilsweb. html (2000)
7. Feldman, J., Monteserin, A., Amandi, A.: Automatic detection of learning styles: state of the art. Artificial Intelligence Review pp. 1–30 (2014)
8. Graf, S., Shuk, K., Liu, T.C.: Identifying learning styles in learning management systems by using indications from students' behaviour. In: Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on. pp. 482–486. IEEE (2008)
9. Hannafin, M.J., Hill, J.R., Land, S.M.: Student-centered learning and interactive multimedia: Status, issues, and implication. Contemporary Education 68(2), 94 (1997)
10. Kolb, A.Y., Kolb, D.A.: Learning styles and learning spaces: Enhancing experiential learning in higher education. Academy of management learning & education 4(2), 193–212 (2005)
11. Latham, A., Crockett, K., McLean, D., Edmonds, B.: A conversational intelligent tutoring system to automatically predict learning styles. Computers & Education 59(1), 95–109 (2012)
12. McNeill, F.M., Thro, E.: Fuzzy logic: a practical approach. Academic Press (2014)
13. Sandholtz, J.H., et al.: Teaching with technology: Creating student-centered classrooms. ERIC (1997)
14. Spring, J.: Education and the rise of the global economy. Routledge (1998)
15. Stephenson, J., Yorke, M.: Capability and quality in higher education. Routledge (2013)
16. Watson, W.R., Watson, S.L.: What are learning management systems, what are they not, and what should they become? TechTrends 51(2), 29 (2007)

17. Yueh, H.P., Hsu, S.: Designing a learning management system to support instruction. Communications of the ACM 51(4), 59–63 (2008)

Behavioral Patterns for Automatic Detection of Learning Styles in Learning Management Systems: a Case Study

Guillermo Salazar Lugo[1], Luis-Felipe Rodríguez[2],
Ramona Imelda García López[1], Adrián Macías Estrada[2],
Moisés Rodríguez Echeverría[2]

[1] Instituto Tecnológico de Sonora, Department of Education, Mexico

[2] Instituto Tecnológico de Sonora, Department of Computer and Design, Mexico

{gsalazar47040, luis.rodriguez, imelda.garcia, adrian.macias, moises.rodriguez}@itson.edu.mx

Abstract. In this paper we present advances of a case study designed to determine which student behaviors are more informative when classifying LMS's users according to their learning style. This case study represents the first step towards developing a mechanism for automatic detection of learning styles in LMS, which takes into account behavioral, affective and performance patterns. The contribution of this paper will benefit researchers and practitioners in the field of educational technology with interest in generating personalized learning environments based in LMS.

Keywords: Intelligent learning environments, behavioral patterns, automatic detection of learning styles, learning management system.

1 Introduction

The education model that prevails in most educational systems is characterized by a method in which the instruction is the same for all learners, without distinction of their particular learning styles and preferences [?, ?]. This traditional education model considers the professor as the main actor of the teaching-learning process, minimizing the role of the learner as an individual that only receives information [?]. Although this model guarantees and facilitates access to education for all individuals, it may hinder the development of skills and learning of most students [?].

For computer science students, the traditional education model represents an important challenge when learning algorithms [?, ?, ?]. The algorithms course requires students to develop analytic and problem solving skills as well as to be able to understand abstract concepts related to the design of algorithms.

Moreover, given that 1) this course is usually offered in the first semester, 2) students may have profiles different from computer science (e.g., accounting or business administration), and 3) each student has a different pace and style of learning, the traditional model may be unsuitable for addressing these issues and helping to develop the required skills. These characteristics of the algorithm course demand a teaching-learning process designed to provide personalized instruction. However, given the multiple constraints in educational institutions (e.g., in terms of infrastructure and human resources), the number of learners that must attend a same course is large, making the generation of personalized learning environments a very complex task.

The generation of learning environments centered on the student can help to address some of the weaknesses of the traditional education model [**?**]. A personalized learning environment is designed to meet the learners' needs, interests, rhythms and styles. To establish a personalized learning environment, it is required the implementation of two mechanisms: 1) a mechanism to understand the situation of the student in terms of emotional and cognitive states, previous knowledge, skills, interests, response to situations related to the teaching-learning process, pace and learning style; and 2) a mechanism to generate personalized learning environments that meet the needs of learners once the mentioned characteristics are identified.

LMS have been successfully used for e-learning [**?**]. This type of educational platform aims at supporting teachers in creating and managing online courses and provide them with a variety of features that can be included in a course such as learning material, quizzes, discussion forums, and assignments [**?**,**?**]. LMS focuses on the presentation of educational material and is a suitable tool for generating personalized learning environments by first implementing a mechanism for automatic detection of learning styles, so that students are characterized, and then implementing a mechanism to adapt the instruction to meet such learning style. The importance of learners' behavior patterns for the automatic detection of learning styles in LMS has also increased in recent years, mainly due to the capabilities of LMS for monitoring and storing data related to the behavior of users (e.g., browsing patterns, time spent on a course, type of resources used).

In this paper we present advances of a case study designed to determine which student behaviors are more informative when classifying LMS's users according to their learning style. This case study represents the first step towards developing a mechanism for automatic detection of learning styles in LMS, which takes into account behavioral, affective and performance patterns. The contribution of this paper will benefit researchers and practitioners in the field of educational technology with interest in generating personalized learning environments based in LMS.

2 Learning Styles and Behavior Patterns

Learning styles describe the manner and the conditions in which learners receive, process, store and retrieve more effectively and efficiently what they are trying to learn [?]. The literature in the field of psychology reports various models of learning styles that have been used in the automatic identification of learning styles [?]. The following list shows the categories used to classify learners in some of these models :

- The Kolb model classifies students into four categories: divergent, convergent, assimilating, and accommodating,
- The Gardner's theory of multiple intelligence defines eight types of intelligence: logical/mathematical, linguistic, spatial, musical, kinesthetic, naturalist, interpersonal and intrapersonal,
- The Felder and Silverman model proposes four dimensions with two styles each: processing (active and reflexive), perception (sensory and intuitive), input (visual, verbal) and understanding (global sequential).

Behavior patterns commonly used in the automatic identification of learning styles are classified into three groups according to the type of information used to infer styles: performance, feedback and behavior. Feldman et al. [?] describe some variables that can be monitored in learning platforms and which can be used to identify learning styles according to the categories defined in the Felder-Silverman model. The following list describes some of these variables organized according to the categories of the Felder-Silverman model:

- Active: number of answered questions and number of performed exercises,
- Reflective: number of visited learning content and number of visits to a forum,
- Sensing: number of right answers given after seeing an example and number of correctly answered questions about details,
- Intuitive: number of right answers given after a theoretical explanation, number of correctly answered questions about concepts, number of correctly answered questions about developing new solutions,
- Visual: number of right answers given after seeing an image and number of images clicked,
- Verbal: number of right answers given after reading a text, number of visits to a forum,
- Sequential: number of times the student chooses to be guided through the steps of solving a problem and number of correctly answered questions about details,
- Global: number of times the student chooses to solve a problem straight away and number of visited outlines.

These variables are also useful to determine the type of user behavior that needs to be monitored in LMS in order to infer learning styles.

3 Proposal

We designed a case study to determine which student behaviors are more informative to classify LMS's users according to their learning style. This case study represents the first step towards developing a mechanism for automatic identification of learning styles. This mechanism attempts to take advantage of common features and functionality in LMS such as Moodle, without making modifications or extensions to the platform.

The case study consisted in preparing academic resources and making them available to learners using a LMS. Then, a classification tool is used for analyzing the user behavior monitored and stored by the LMS and the data obtained from applying a learning style instrument to participants. This analysis process results in the definition of behavioral patterns useful to identify user's learning styles. Figure 1 illustrates the elements included in the proposed case study and their interrelationships.

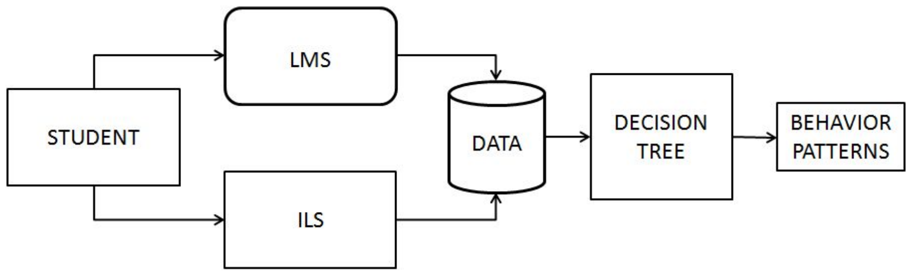


Fig. 1. Components included in the case study proposed

In the case study participated 134 students enrolled in an introductory course of computational Algorithms. This course is aimed at first semester students of Software Engineering at the Technological Institute of Sonora. All students used Moodle during five days as a support for classroom instruction. The instructional role of Moodle was to reinforce the topics reviewed each day. Moodle is an open source LMS, which allows the modification of the system to meet different needs. Moodle includes mechanisms to show information of learners in terms of number of access to resources and activities of courses, their qualifications, participation in groups, forums, chats and others. This information allows the user and professors to understand the learners' behavior within the platform. Based on these advantages of Moodle and the fact that it is highly adopted by the instructional community, this educational platform was selected as the LMS to be used in the proposed case study.

The academic resources prepared for learners are related to topics of the Algorithm course. These resources are in various formats such as lectures, videos, and presentations. Resources and Activities included in Moodle and presented to the students were used to obtain information about their behavior. Moreover, the task component was used to design exercises while the page component for designing learning content, examples, and outlines. We prepared different versions of each type of resource in terms of complexity. For example, exercises were classified as basic, medium and advanced. Moreover, each learning content were classified in text, graphic or video. Chats and forums where also used for learners' communication and collaboration. All these features are common to most LMS, including Sakai, Claroline, and RLN.

Once students finished the course, data related to their behavior were extracted from the Moodle database and combined with the results of the ILS questionnaire to generate a consolidated database which will serve as a data source for Weka, a tool that facilitates the data analysis using well-know algorithms. In particular, we will use Weka to generate a learning decision tree in order to identify patterns of behavior that are informative to infer users' learning style.

4 Preliminary Results

The following list describes the steps that have been carried out according to the case study described in the previous version:

1. Setting up the LMS. Moodle was installed and configured on a server with public IP to provide access to students via the Internet. Students were provided with a user-name and password.
2. Course and Content. A new course was configured in Moodle and four modules included (one for each topic of the introductory algorithms course). The course content was composed by three basic exercises, three intermediate exercises, three advanced exercises, four assignments, four forums for discussing learners' questions and contributions, two outline pages, sixteen examples, six graphic learning content, four text learning content and three video learning content (see Figure 2).
3. Students Style Identification. Students were asked to answer the Spanish version of the Index of Learning Style (ILS) questionnaire based on the Felder-Silverman model (available online) [**?**]. The first day of the course, students sent the completed questionnaire and the results page generated by the site (the learning style of each student). The information of each student was then captured in the Moodle database.
4. Behavior monitoring. During five days, students used Moodle to send assignments and reinforce their knowledge on each topic using the learning contents, exercises, examples, forums and chat rooms freely without it being mandatory in any of the cases. To record the student activity, preset options of event logging in Moodle were also used.
5. Data extracting. SQL queries on tables containing information related to student behavior within the platform were generated. The SQL queries were designed to answer the following questions for each student:
   – How many exercises does the student visited?
   – How many exercises visited by the student are basic, intermediate and advanced?
   – How many learning contents visited by the student are text, graphics and video?

   – How many times the student visited a forum?
   – How many times did the student participate in a forum?
   – How many times did the student visit the outlines?
   – How many times did the student visit examples?
   – How many times did the student visit a chat?
   – How many times did the student participate in a chat?
   – How many times did the student login Moodle and at what time?

6. Data analysis. Students were classified in terms of their learning style based on the score obtained in the ILS questionnaire, which considers the following categories of the Felder-Silverman model: active / reflective, sensitive / intuitive, visual / verbal, sequential / global. Participants were discarded when their learning style was balanced for all dimensions or when their scores in two or more categories were high. Students with a clear bias towards a specific style were easily classified. For example, if a student had the classification values ACT_REF=equilibrated, SEN_INT=highly intuitive, VIS_VRB=moderated visual, SEQ_GLO=moderated global, then the style assigned to this student was *intuitive*. However, if a student had the classification values ACT_REF=equilibrated, SEN_INT=equilibrated, VIS_VRB=moderated visual, SEQ_GLO=moderated global, then it was considered that there was not enough evidence to clearly define a learning style (in this case the data were discarded). Table 1 shows the classification criteria used for each dimension of the Felder-Silverman model.

Table 1. Classification criteria by dimension.

Dimensions	**Classification Values**				
ACT_REF	Highly Active	Moderated Active	Balanced	Moderated Reflexive	Highly Reflexive
SEN_INT	Highly Sensitive	Moderated Sensitive	Balanced	Moderated Intuitive	Highly Intuitive
VIS_VRB	Highly Visual	Moderated Visual	Balanced	Moderated Verbal	Highly Verbal
SEQ_GLO	Highly Sequential	Moderated Sequential	Balanced	Moderated Global	Highly Global

From 134 students enrolled in the course, only 82 answered the ILS questionnaire and after data analysis, 55 students were correctly classified into a learning style as shown in Table 2.

5 Conclusions and Future Work

In this paper we proposed a case study to determine behavioral patterns for automatic detection of Learning Styles in LMS. We presented and described the

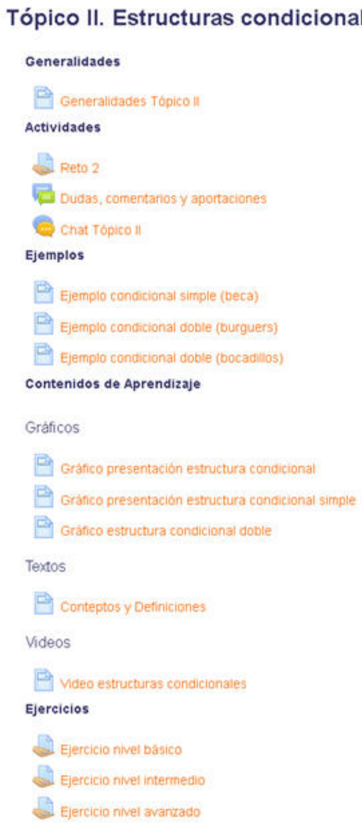


Fig. 2. Interface of the LMS including academic material

main components included in this case study as well as phases that were carried out. Preliminary data extracted from the Moodle database provide evidence for the feasibility to monitor and record data related to the behavior of students in an open license LMS without having to carry out special configurations or functionality extensions. As future work, the data generated in this preliminary work will be used as input to Weka in order to generate a learning decision tree. We attempt to employ this technique to determine which user behaviors in a LMS are more informative when classifying students according to their learning style. Once behavior patterns are associated to each learning style category, we will carry out the proposed case study but omitting the application of the ILS instrument in order to validate the results. We are also planning to monitor the dynamics of students in terms of their affective state when using the LMS. The main purpose is to understand how informative are affective aspects for the automatic identification of learning styles.

Table 2. Learners classified according to their learning style

Learning Style	Frequency	Percentage
Active	12	22
Reflexive	6	11
Sensitive	5	9
Intuitive	5	9
Visual	18	33
Verbal	3	5
Sequential	3	5
Global	3	5
Total	**55**	**100**

Acknowledgments. This work was partially funded by the SEP-PRODEP through the project ITSON-PTC-089.

References

1. Arnove, R.F.: Comparative education and world-systems analysis. Comparative Education Review pp. 48–62 (1980)
2. Darling-Hammond, L.: Teacher learning that supports student learning. Teaching for intelligence 2, 91–100 (2008)
3. Despotović-Zrakić, M., Marković, A., Bogdanović, Z., Barać, D., Krčo, S.: Providing adaptivity in moodle lms courses. Journal of Educational Technology & Society 15(1), 326–338 (2012)
4. Feldman, J., Monteserin, A., Amandi, A.: Automatic detection of learning styles: state of the art. Artificial Intelligence Review pp. 1–30 (2014)
5. Gomes, A., Mendes, A.J.: Learning to program-difficulties and solutions. In: International Conference on Engineering Education–ICEE. vol. 2007 (2007)
6. Graf, S., Shuk, K., Liu, T.C.: Identifying learning styles in learning management systems by using indications from students' behaviour. In: Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on. pp. 482–486. IEEE (2008)
7. Hannafin, M.J., Hill, J.R., Land, S.M.: Student-centered learning and interactive multimedia: Status, issues, and implication. Contemporary Education 68(2), 94 (1997)
8. Holzman, L.: Schools for growth: Radical alternatives to current educational models. Lawrence Erlbaum Associates Publishers (1997)
9. James, W.B., Blank, W.E.: Review and critique of available learning-style instruments for adults. New Directions for Adult and Continuing Education 1993(59), 47–57 (1993)
10. Jenkins, T.: On the difficulty of learning to program. In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences. vol. 4, pp. 53–58 (2002)
11. Llorente Cejudo, M.C.: Towards e-learning from free software. moodle like a learning managament system (lms) within reach of all. COMUNICAR (28), 197–202 (2007)

12. Moroni, N., Señas, P.: Estrategias para la enseñanza de la programación. In: I Jornadas de Educación en Informática y TICs en Argentina (2005)
13. Solomon, B.A., Felder, R.M.: Index of learning styles. Raleigh, NC: North Carolina State University. Available online (1999)

Design of Multi-Agent System for Solution of the School Timetabling Problem

César Covantes, René Rodríguez

Universidad Autónoma de Sinaloa,
Facultad de Informática Culiacán, Culiacán, Sinaloa,
Mexico

c.covantes11@info.uas.edu.mx, rene.rodriguez@info.uas.edu.mx

Abstract. This paper presents the analysis and design of a system based on *multi-agent systems* (MAS) by negotiation with JADE framework (Java Agent DEvelopment) to solve the school *timetabling problem*. In the design, the system considers three types of agents; A coordinator agent responsible for instantiate, create and manage the group agents, where the number of teacher and group agents depends on each case study. The group agents perform the negotiation in order to solve the conflicts between all the teacher agents. The system takes the time, space, activities and other type of constraints by FET (Free Timetabling Software) format in an XML and to prove the algorithm were considered for the analysis and experimentation the case studies Belize, Brazil, Spain and UK.

Keywords: Agent, multi-agent system, timetabling, xml, objective function, FIPA.

1 Introduction

Humans on daily basis plan what activities do in a day and in a certain period of time, but the problem is to select, assign resources and time to obtain a set of activities in an organized manner, resulting a schedule where the order and completion time is important. The persons who performs the schedules should consider different factors such as priorities, time to devoted activities, space availability, cost and valuation of the consequences, satisfying a set of hard and soft constraints [19]; the hard constraints are actions that must satisfy all circumstances, while soft constraints represent a greater flexibility, can be able to satisfy or not, reflecting a temporal relation between activities, given the limited capacity of shared resources.

A common problem behind these assignments is the problem of *timetabling problem (TTP)*. In the area of computer science, the *timetabling* represent an optimization problem that belongs to the family of the NP (non-deterministic polynomial time) problems [19]. The NP problems have a computational complexity with a large space search or combine all possible solutions to a problem, where the goal is to find "good" solutions by an evaluation function that describes its quality in an "acceptable" time.

In some cases, the problem is formulated as a *search problem*, trying to find a schedule satisfying all the restrictions (hard and soft), while in other cases, the problem is formulated as an *optimization problem*, trying to find a schedule that satisfies all the constraints hard and minimize (or maximise as appropriate) through an objective function the soft constraint, applying optimization techniques to a search problem.

This type of problem is only permitted only for a small number of cases (*e.g.*, less than 10 courses) [19], whereas real instances usually may involve a few hundreds of courses. The problem is still present, even though there are different methods that have been developed and used to solve the *timetabling problem* on specifics departments and institutions which are not universal methods so a proposed solution cannot solve "any TTP" problem [20].

The different proposed methods to solve the timetabling are rarely compared with each other by the lot of number of different variables and different ways of quantifying the constraints raised from different policies and practices, which each has its particular characteristics courses assignment. The comparison is necessary to determine what is or are the best computational methods given the different types of data schedules, allowing discard simple techniques, through a manner in which the information it is represented and exchange and unify the restrictions given the different institutions. The eXtensible Markup Language (XML) is used as a standard for data storage, making it useful for several applications that communicate with each other, in addition to exchanging information between different platforms.

2 Theoretical Framework

2.1 School Timetabling Problem

In the school timetabling problem there are "participants" in a fairly general sense, *i.e.* teachers, classes, lecture halls, laboratories, pieces of equipment, and so on. In addition, there exists a set of "hours", sometimes called time slots or periods. The term "availabilities" describe for every participant the subset of hours in which he (it) is free, willing or able to participate in one of the lessons, lectures, conferences or examinations in which he (it) is involved. The latter events are subsumed under the notion of a meet. Every "meet" is described by the collection of participants which have to come together and by the number of hours required for it [20]. The class-teacher timetable problem is obtained if every meet keeps busy exactly one teacher and one class as participants. However, there may be a demand for a meet, which consists of a gymnastics lesson to be held by a male teacher and a female teacher each in different gymnastic hall at the same time. Finally, there may be preassignments of some meets to hours.

Given such a situation, a timetable is a schedule assigning to all these meets the precise number of hours required, so these hours are available for all participants of the meets and such that, as a fundamental requirement, none of the participants is scheduled twice in the same hour.

However, there is a diversity of special requirements a timetable must observe depending heavily on the type of school and on administrative peculiarities of the country [20]. If not, can be problems, is why the problem should consider the following recommendations: consider the division of the set of hours into days if the scheduling cycle is the week. For some participants (namely the classes), it is necessary to avoid free hours between other lessons; they may have free hours only at the beginning or the end of a day. Some subjects require consecutive hours not straddled by a break. There may be limitations on a teacher's daily load, and it may be necessary to provide every teacher with a free day. Subjects taught several times a week should be spread evenly throughout the week. Teachers may indicate a preference on the length of the interval between their lessons. Of course, not all of their claims are equally important; some are merely aesthetic constraints (soft).

The requirements between school levels are different. While in schools, the size of a class is of minor interest, it becomes important at universities, because the number of students in a lecture may vary [20]. In universities the rooms can be selected from a set of rooms of comparable size, while schools each class can be busy all the time by the same teacher. On the other hand, the requirement for the distribution of free hours over the week of either students of lectures are far less restrictive.

There are a large number of variants of the timetabling problem, which differ from each other on the type of institution involved (university or school) and the type of constraint. Therefore [19] classify the timetabling problem into three main classes:

- School timetabling,
- Course timetabling,
- Examination timetabling.

In *school timetabling* the scheduling is weekly for all the classes at school, avoiding two teachers meeting classes at the same time, and vice versa, while the *course timetabling*, the scheduling is weekly for all the lectures of a set of university courses, minimizing the overlaps of lectures of courses with common students. The *examination timetabling*, the exams scheduling is for a set of university courses, avoiding the exams course overlapping with common students, and spreading the exams for the students as much as possible. Such classification is not strict, some problems can fall between two classes, and cannot be easily placed within the above classification [19].

2.2 Distributed Approaches

This section provides a review of work with MAS, in the literature there are a large set of optimization techniques to solve the TTP from the appearance of meta-heuristics in 1983 [21]. In [7], proposed a resolution with multi-agent once the schedule is built, presenting problems of allocating the groups when they are moving from one room to another. A year later[5] arise a model to decompose the secondary scheduling problem into sub-problems and solve each sub-problem in parallel by a decomposition algorithm used to divide a graph in sub-graphs running on a different machines.

After that, in [25, 26] presents two works with MAS. The first, used two mobile agents whose behavior was to verify daily conflicts, while the second work, the agents represent the hard constraint. In [11] presents a hierarchical approach combining a small recursively form in a large one. Also, in [12] presents the problem of school TTP using the model of parallel processing using 'coarse grained' facilitating the exchange of cases and the creation of multiple solutions in parallel. Finally, [15] present the problem *course timetabling* with MAS from a distributed approach.

In all works with MAS, are limited in the specifications of the restrictions and the way which the agents resolve the conflicts, also the works doesn't have a good abstraction and implementation of the problem lacking of information to be implemented in other case studies, being then particular implementations.

2.3 Problem Representation

The unavailability interchangeable TTP benchmarks in a uniform format to express different sets of data and facilitate the public use of these problems, until in [18] UniLang language is presented to define school, course, examination in a language similar to XML format, presenting a tester to validate if a scheduling satisfy with all the requirements and limitations defined by the problem.

In [10] is created the language called STTP (School TimeTabling Problem) with the same XML structure to specify the TTP for high schools and evaluate solutions to these problems. Then, in [17] takes a similar approach to XML format with his own specifications and structure called XHSTT (Xml High School TimeTabling) to facilitate the exchange of data and promote the research in this area for high school with 16 different case studies of different countries located in a repository for a public access with an evaluator.

In [4] an XML format is presented with his own structure and extension called FET with 22 case studies of different countries, located in a repository for a free access. FET has specifically different types of restrictions to their representation in addition used as a tool to capture the problem.

2.4 Existing Software

Nowadays exist software commencial and free that tries to solve the TTP. Every software uses their own optimization techniques and structure restrictions, while others only are a support tool for the allocation of schedules manually. Within the commercial software are aSCTimeTables and Mimosa; aSCTimeTables [1] allows the automatic scheduling which does not specify the search method and the restrictions. The Mimosa software [14] allows the automatic scheduling for schools, meetings, courses and conferences planning without specifying the solution method and restrictions.

Moreover in free software are FET, iMagic, Mimosa 12 Scheduling Software, openSIS, TimeFinder, Lantiv and University Timetabling Comprehensive Academic Timetabling Solution; FET [4] is a GNU license which allows the scheduling automatic in particle swarm, specifying also its structure of the problem and restriction in XML format. iMagic [8] allows the TTP automatically without specifying the search method for a solution and a restrictions. Mimosa 12 Scheduling Software [14], has also a free version with a solution automatically without specifying the solution method and format restrictions. openSIS [16] allows the TTP, also as a tool to fill the grades and inscriptions information without specifying the solution method and format restrictions. TimeFinder [22] uses an optimization algorithm which is not specified as well as the restrictions. Lantiv [13] is a software that serves as a tool for programming schedules manually without specifying the structure of the restrictions. Instead, University Timetabling Comprehensive Academic Timetabling Solution[23] is open source with GNU license that allows the scheduling of examinations and uses minimizing conflicts with local search using constraint programming (variables, values and limitations) without specifying the format restrictions.

3 Multi-Agent System

A Multi-Agent System (MAS) is a system in which several agents interact and pursue a set of goals or tasks to achieve a goal [24]. The agent's features have utility function on a set of goals. Among the techniques that the utility function agents have is to increase it in the points below:

- Coordination,
- Cooperation,
- Directed behavior,
- Planning,
- Communication.

3.1 Agent Platform

Exist an organization called The Foundation for Intelligent Physical Agents (FIPA) responsible to produce software standards specifications for heterogeneous and interacting agents and agent based systems. In the elaboration of those standards, it is used for interoperability between utility-based agent developed by different companies and organizations. The FIPA organization belongs to the IEEE Computer Society standards for their standards, interoperability and software development that promote with the agent technology and other technologies [5].

The agent platform (AP) provides the physical infrastructure in which agents can be deployed. The AP consists in machines(s), operating system, agent support software, FIPA agent management components (DF, AMS and MTS) and agents.

3.2 Agent Management Reference Model

Agent management provides the normative framework within which FIPA agent exists and operates. It establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agent [6].

The agent management considered an AP which provides the physical infrastructure in which agents can be deployed. The AP consists of four components called Agent, Agent Management System (AMS), Directory Facilitator (DF) and Message Transport Service (MTS). The first three components are especial types of agent that support and management of agent, while the MTS provide a delivery message service. The function of this element is located in the specification on web site of FIPA [6].

The internal design of an AP is an issue for agent system developers and is not a subject of standardization within FIPA and the entities contained in the reference model (Fig. 1), between then: external software, agent, agent management system, directory facilitator and message transport service.

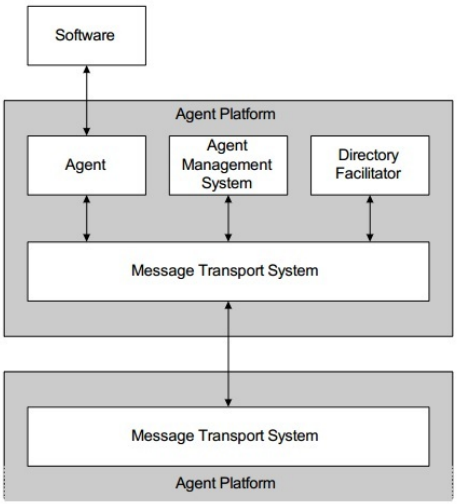


Fig. 1. Reference architecture of a FIPA Agent Platform.

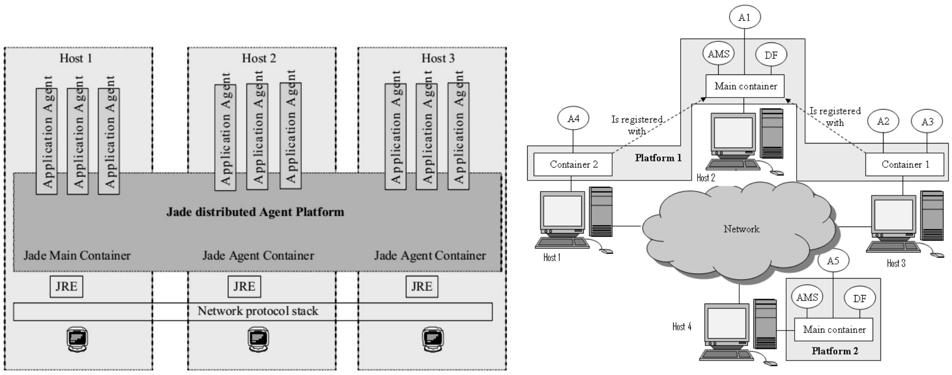
3.3 Agent Platform Implementation

An AP provides the physical infrastructure in which agents can be deployed for archives their goals. FIPA presents a list of mayor implementation of MAS development with public access [5].

Of the 11 platforms that meet the standards of FIPA, JADE was considered because has the advantage of being updated [9] unlike April, FIPA-OS, Grasshopper, Java Agent Services API, LEAD and ZEUS. Also, JADE is free and Open

Source unlike CAPNET and JACK that need license. JADE present security features for authentication connections, user validation and message encryption. Besides, presents a complete graphical interface, great documentation and high acceptance in companies, scientific community and development projects.

In addition [9] has the advantage of being able to distribute into different containers or equipment in a remote mode in order to reduce the number of threads per host on different computers even though they don't have the same operating system. Besides, another feature of JADE is that can be managed through a graphical interface for the communication between platforms. JADE system is made by one or more agent containers, each one with different java virtual machine (Fig. 4).



A) JADE distributed 1. B) JADE distributed 2.

Fig. 2. JADE Agent Platform distributed over several containers.

4 Methodology

4.1 Constraints Representation

This section describes the structure and design of FET. First FET has the advantages of being available, defined structure, easily public access, many case studies from different countries presented, besides being portable with many constraints divided into four groups shown below:

- Time,
- Space,
- Activities,
- Other type.

The 'time constraints' have the constraint 'not available' that correspond to the day time where teacher or group mustn't be an activity assigned. The constraint 'max days per week' are the days allowed per week that a teacher could teach a class, several teachers, a group or several groups for activities. The 'max gaps per week or per day' are an unused timeslot or several between two activities. The constraints 'max or min hours daily' are the hours or groups can have daily and finally 'max hours continuously' are the hours that can be assigned to a specific teacher or group or all the teachers or groups belonging to the institution, this constraint can be used to affect the minimum number of gaps per day.

In the space constraints 'Home room' is when a teacher, teachers, group or several groups have a default room or rooms. While 'Max building changes per day or week' are the times they have permitted to move between buildings per day o per week and finally the restriction 'Min gaps between building changes' are an unused timeslot or several between two building changes.

In order to know the activities restriction it is necessary to know how an activity is made. An activity are made by a teacher, subject, students (group), duration, total duration, an identifier (id), and identifier of group activity, active and comment, which a teacher has a set of activities that are linked with the corresponding group, so a set of activities make up a schedule.

Given the attributes of an activity, it's possible to assign them either individual or in a group, within the 'activities constraints', the constraint 'An activity has a preferred starting time' is when one activity have a special period to be assigned, moreover the constraint 'An activity has a set of preferred starting time' correspond the set of activities preferred to be assigned. The slot constraint have 'An activity has a set of preferred time slots' is when one activity has a set preferred slot to be assigned. The constraint 'A set of activities has a set of preferred time slots' correspond a set of activities. The constraint 'Min n days between a set of activities' is when a set of activities should be instructed on different day. The constraint 'An activity ends students day' is the activity that mush end a student's day and that activity have a common attribute like a special subject, while the constraint 'A set of activities ends students day' correspond a set of activities. The activity 'A set of activities has same starting time' is when a set of activities should be starting in the same day an hour, while the constraint 'A set of activities has same starting day' is only for a day and the constraint 'A set of activities has same starting hour' is only for an hour.

Also the constraint '2 activities consecutive' are activities that are one after each other as opposed to '2 activities are ordered' which the activities can be assigned in the morning and the other in the afternoon no matter other activities is or are in the middle. The constraint 'Min gaps between a set of activities' are the slot unassigned to allow mobility group and finally 'A set of activities are not overlapping' the activities never be assigned in the same slot.

The 'other restrictions' not correspond any of the above classifications but are important, within them are the 'Basic compulsory time constraints', 'Basic compulsory space constraints', 'Break', 'A room is not available' and 'Preferred room(s)'. The constraint 'Basic compulsory time constraints' a teacher never has to instruct two or more activities at the same time, also students must have maximum one activity per period. In the 'Basic compulsory space constraints' the rooms will never include two or more activities. The constraint 'Break' is the way to say that all teachers and students are not available regularly to indicate a break for lunch and finally 'A room is not available' for a subject, group, teacher or activity.

4.2 Multi-agent System Design

In the system design of multi-agent platform for the solution of *timetabling problem* the agents/actors are created in 3 different stages throughout the system (Fig. 3). The stages of multi-agent system are explained in detail in the following sub-paragraphs.

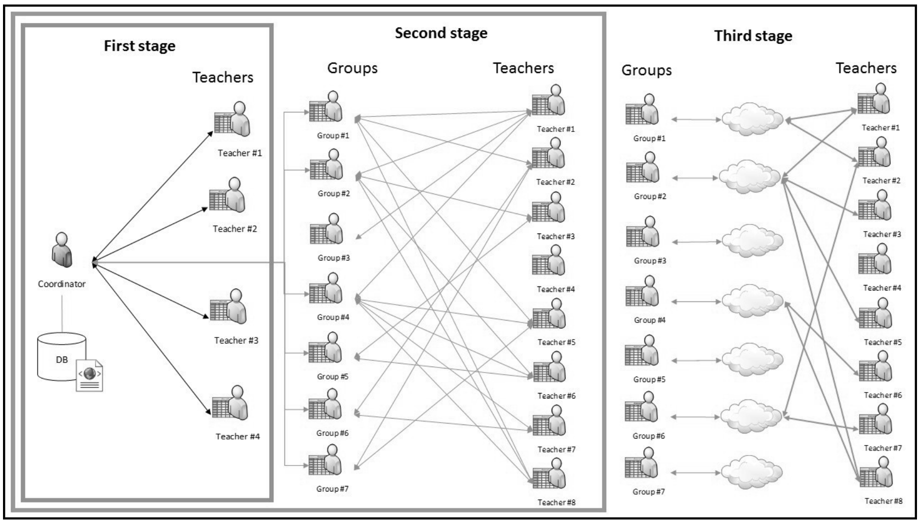


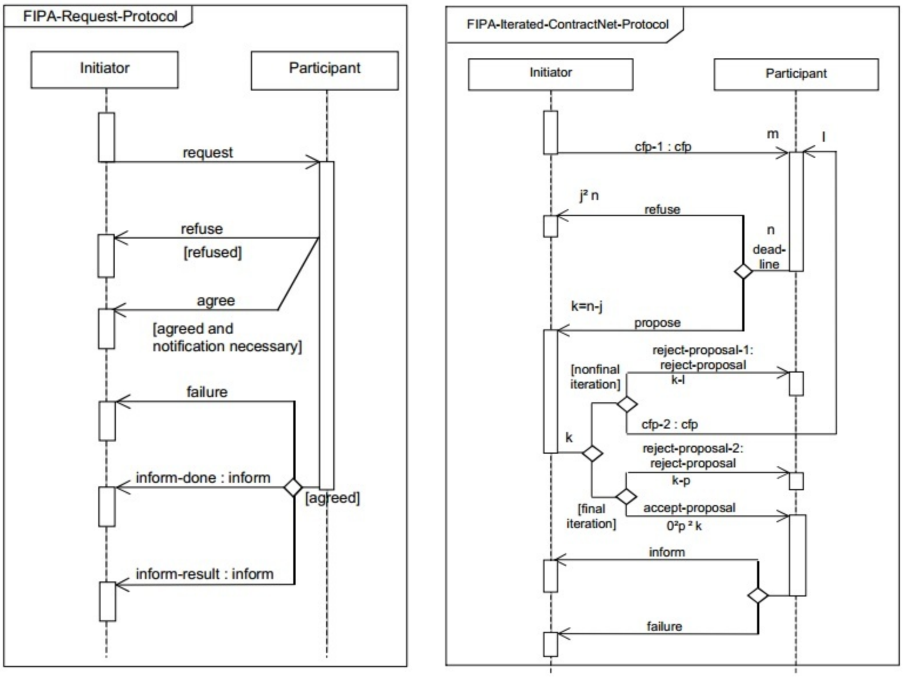
Fig. 3. Design of multi-agent system (MAS).

4.3 Negotiation Strategy Design: First Stage

In this first stage a coordinator agent is created, register with the DF and is the responsible for manage the agents teachers and groups. The coordinator take the activities constraints through an XML file that contain the information of the institution of the case study, also the coordinator agent read the file and create the agents teachers who are registered with the DF, their obtain their constraints and activities, then based on their activities, perform their selfish proposal individually according to their preferences, as long as their respect the general constraints of the institution but respecting part of the schedule suggested by the teacher. The agent professor when finished making the schedule, notify the coordinator.

4.4 Negotiation Strategy Design: Second Stage

In this second stage when all the teachers are in status ready, the coordinator agent creates the agents groups, and each group is register with the DF, the groups obtain the restrictions, activities and the request the schedule of every teacher using the FIPA-request protocol (Fig. 4.A), being the initiator the group on request the teacher schedule and the participant the teacher sending by a response his/her schedule.



A) FIPA request protocol - Second stage.   B) Iterated Contract - Net Protocol.

Fig. 4. FIPA Protocols.

4.5 Negotiation Strategy Design: Third Stage

Once all teachers have sent their schedule to the agents group, the agents group verifies if exist a conflict; if exist with a day and hour, the group sends the schedule with the available space to all the teachers involved in the conflict to propose a new position until the agents teachers does not present any problem to move or when they respect the largest number of restriction by an evaluation function. The teachers send the proposal requested by the agent group who evaluates all the proposals received in order to accept/reject the proposals or if is necessary request another position (day and hour) to the agent teacher if the space has already been occupied (assigned) for another negotiation that perform in parallel because exist several negotiations and wins who evaluate and assign first.

The above, for being a distributed approach where all groups are trying to resolve conflicts simultaneously with all teachers, there are cases where a teacher has a conflict with other teachers in several hours with different groups and the same teacher send the same position for different groups, so the group validates and if it was assigned the agent group request a new negotiation to the teacher to provide a new position. The implemented negotiation protocol is the iterated contract-net where de Initiator is the agent group and the Participants the teachers (see Fig. 4.B).

In contrast to contract-net protocol, the iterated contract-net protocol allows new negotiation rounds and is useful when there are cases as mentioned above with several negotiations in occupied spaces, and then the group request each teacher to send another proposal. The way in which the agent group accepts or rejects the proposals of teacher's agents will be explained below with six possible scenarios with three teachers as show in Table 1.

Table 1. Scenarios with three teachers.

Scenario 1	**Scenario 2**	**Scenario 3**
Teacher 1: 0	Teacher 1: 100	Teacher 1: 100
Teacher 2: 0	Teacher 2: 0	Teacher 2: 100
Teacher 3: 0	Teacher 3: 0	Teacher 3: 100
Scenario 4	**Scenario 5**	**Scenario 6**
Teacher 1: 100	Teacher 1: MAX	Teacher 1: MAX
Teacher 2: 200	Teacher 2: MAX	Teacher 2: MAX
Teacher 3: 300	Teacher 3: 300	Teacher 3: MAX

In an evaluation of conflicts that makes a group in the scenario 1, the three teachers no present a problem to move to another space, so the agent group cancel the request for only one teacher randomly, while others accept and change them. In scenario 2 the teacher 1 can be change for a new position but not respecting one or more constraints with a weight of 100 as a result of an evaluation function while the others teachers have no problem to move. The group cancels the request of professor 1 and accept the new position the teachers 2 and 3. In the scenario 3 due all the teachers have problems in enforcing restrictions with the same weight, the group performs the same procedure in the stage 1, cancel one at random and accept the rest teachers.

In scenario 4, the group cancel the request of professor 3 because present the major problem to move, while the others the group accepts them his new position. In stage 5, the teachers 1 and 2 have a maximum value implying that they didn't find an available position, which whereby the group cancels one of the professors 1 and 2 at random and only accepts the teacher 3, and finally on stage 6 none teacher found a new position which the group accepts the teacher request without making changes.

5 Experiments and Results

5.1 Selection of Case Studies

This section describes the experiments and results obtained by implementing the algorithm based in multi-agent system negotiation to resolve conflicts between teachers. In this experimentation process they were considered four case studies at random from a 22 countries. These case studies are freely available in a repository by FET [4].

The countries considered for the experimentation are; Belize, Brazil, Spain and UK, which each one have a specific characteristics in hours, day, groups, teachers, subjects, and activities. The activities of Belize and Spain were modified because there were null data, Belize with a total 952 were 249 null data, while Spain a total of 1086 activities were found 269 data null, giving as a result the data shown in the table 2.

The countries considered for the experimentation are; Belize, Brazil, Spain and UK, which each one have a specific characteristics in hours, day, groups, teachers, subjects, and activities. The activities of Belize and Spain were modified because there were null data, Belize with a total 952 were 249 null data, while Spain a total of 1086 activities were found 269 data null, giving as a result the data shown in the table 2.

Table 2. Characteristics of the case studies.

	Belize	**Brazil**	**Spain**	**UK**
Days	5	5	5	6
Hours	6	5	7	5
Groups	23	16	185	46
Professors	44	27	56	26
Subjects	24	12	78	25
Activities	703	400	817	163

Within results also was develop a system to validate the proposed model, in the Fig. 6 shown the interface execution in the case of Brazil and so for each of the selected cases, in order to see the details on the development (cf. [2]). In this execution is shown the agent coordinator; the first column indicates the name of the agent (agent), the second the type of agent (Type of Agent) in case of teacher or group and finally the agent status (Status). The Fig. 6 refers to the first stage that was shown in Fig. 3.

Also, to validate the second stage of the model in Fig. 3, the system shown in Fig. 6 was implemented once all teachers are ready in status 'Ready', the user

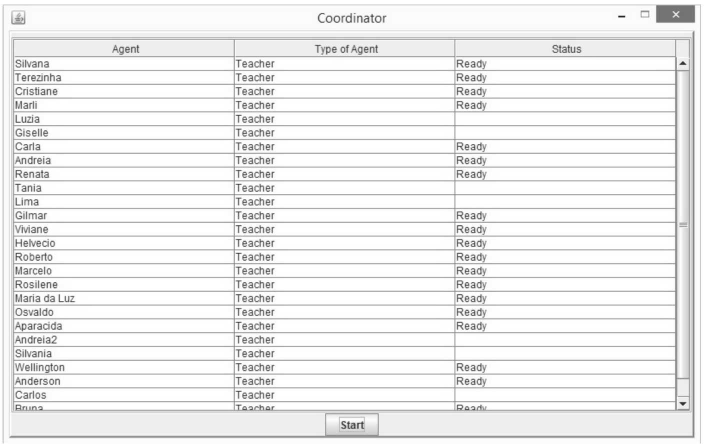


Fig. 5. Screenshot coordinator GUI with teachers.

can press the button *Start*. The teachers send the proposals to the groups and the groups check the spaces where exist conflicts to start the part 3 of negotiation between the teachers involved.

5.2 Conflict Resolution

In the experiment, for each case study were considered 30 executions in order to identify the average of the conflicts presented by each group as well as the unresolved conflict, the initiated protocols and the number of message between agents. The table 3 shows the results of case studies.

Table 3. Results of case studies.

	Found conflicts			Resolved conflicts			Initiated protocols			Messages		
	Total	Average	SD	Total	Average	SD	Total	Average	SD	Total	Average	SD
Belize	3752	125.07	6.02	3505	116.83	5.25	11340	378	8.26	50845	1694.83	44.74
Brazil	4149	138.30	7.30	3067	102.23	5.72	11095	369.83	8.67	51369	1712.30	49.77
Spain	369	12.3	2.83	369	12.3	2.83	13800	460	3.29	42865	1428.83	22.05
UK	3	0.1	0.31	3	0.1	0.31	7083	236.1	0.31	21260	708.67	2.04

The results of Belize the average have 125.07 ($\pm$ 6.02), are resolved 116.83 ($\pm$ 5.25) and 8.23 ($\pm$ 3.18) are unresolved, while the initiated protocols the average is 378 ($\pm$ 8.26) and messages between agents 1694.83 ($\pm$ 44.74). In the results of Brazil the average have 138.3 ($\pm$ 7.3) conflicts found in a initial way, 102.23 ($\pm$ 5.72) are resolved with the algorithm and not resolve by the algorithm 36.07 ($\pm$ 6.83), while the average in the initial protocols are 369.83 ($\pm$ 8.67) and the exchange message between agents are 1712.3 ($\pm$ 49.77). The results of Spain show

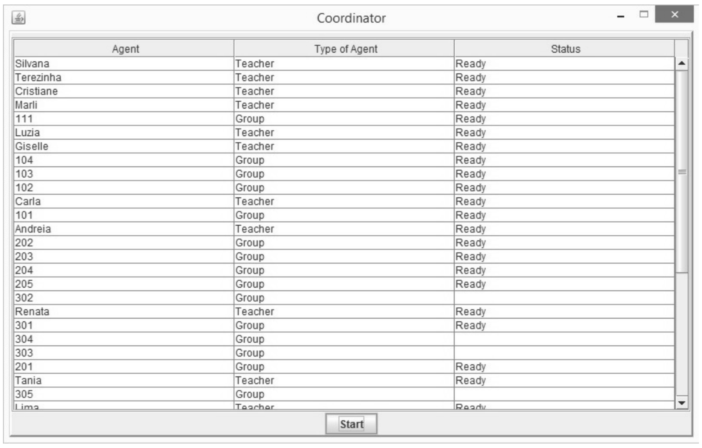
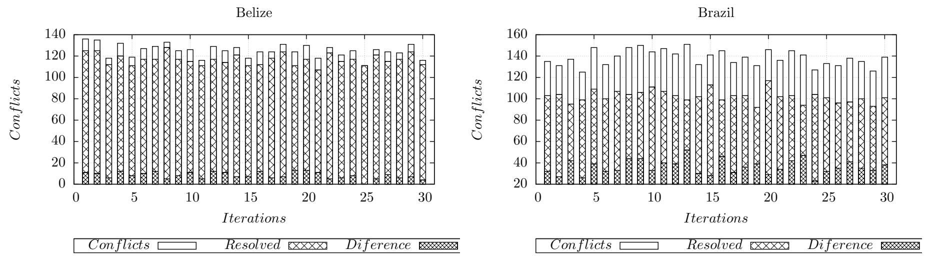


Fig. 6. Screenshot coordinator GUI with teachers and groups.

the found conflicts and resolved with the same average 12.3 ($\pm$ 2.83), taking a little variability between the conflicts fully resolved with the algorithm, while in the case of UK only 3 iterations showed conflicts which could solve having an average 0.1 ($\pm$ 0.31).

The following figures show the result of table 3 by iteration according to conflicts found, resolved and could not solve it. In Fig. 7.A the result shown Belize, while in Fig. 7.B the results of Brazil.

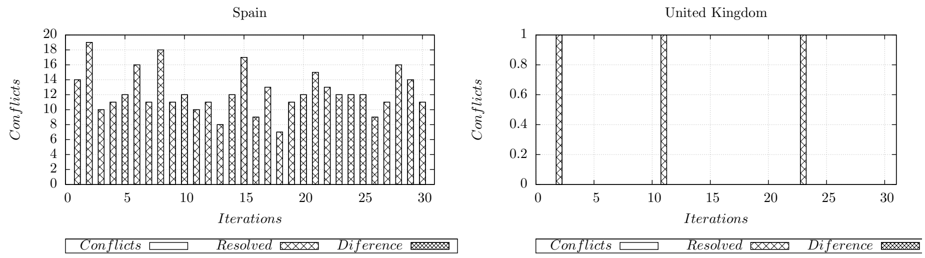


A) Belize resolved conflicts. B) Brazil resolved conflicts.

Fig. 7. Belize and Brazil resolved conflicts.

The Fig. 8.A shows the results of the case study Spain, which shows the overlapping results from the conflicts because the algorithm could solve in fully the conflicts, while Fig. 8.B shows the results of the case UK with only 3 conflicts that could be resolved with a difference of zero having no conflicts in other iterations.

The graphs show the variability of conflicts for cases of Belize and Brazil, the



A) Spain resolved conflicts. B) UK resolved conflicts.

Fig. 8. Spain and UK resolved conflicts.

graphs behave similar because its characteristics has less space for the allocation the activities of teachers in contrast to Spain and UK, also in conflicts found in Belize and Brazil the teachers from an initial state have a schedule that more adapt his/her convenience, imposing more restrictions, while in Spain and UK the teachers impose fewer restrictions, resulting in greater variability in assignment activities.

6 Conclusions and Future Work

The contributions in this work is a system that implements a multi-agent system through negotiation to resolve the *school timetabling problem* allowing the incorporation of general constraints to model different case studies, regardless of the institution or country that want to solve as long as comply with the restrictions and standards of FET.

Another contribution of this work is to allow the agents to interact and solve conflicts in a dynamic way and not deterministic, also one of the advantages of the proposal is that it is compatible with FIPA specifications to implement recognized standards, as well as present regardless of the operating system. In the part of the system like advantage to implement multi-agent system it was possible to consider the initial proposals of teachers, be a difficult task for other techniques that have to generate solutions and validating every assignment.

Based on the result shown in tables and graphs, it's clear that multi-agent systems can in great measure resolve the conflicts; solving in the case of Belize 93.41%, also with Brazil 73.92%, whereas Spain 100% and UK 100%.

As a part of the future work it is necessary to implement the missing constraints. Validate the results obtained with Multi-Agent Systems and compare the solutions generated by FET that implements a particle swarm technique.

Another point to be addressed in future research will implement a new technique to solve conflicts that they could not resolved through a similar technique proposed by researchers [3] through Eco-Problem-Solving.

References

1. Consultants, a.A.S.: aSc timetables. http://www.asctimetables.com/timetables_en.html (2013)
2. Covantes, C.: Sistemas Multi-Agentes para la solución del problema de programación de horarios escolares mediante negociación. Master's thesis, Universidad Autónoma de Sinaloa (2014)
3. Drogoul, A., Dubreuil, C.: A distributed approach to n-puzzle solving (1993)
4. FET: FET free timetabling software. http://lalescu.ro/liviu/fet/ (2013)
5. FIPA: The foundation for intelligent physical agents. http://www.fipa.org/index.html (2012)
6. FIPA AMS: FIPA agent management specification. http://www.fipa.org/specs/fipa00023/XC00023H.html (June 2001)
7. Gaspero, L.D., Mizzaro, S., Schaerf, A.: A multiagent architecture for distributed course timetabling. In: Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling. pp. 471–474 (2004)
8. iMagic Software: iMagic timetable master. http://www.imagictimetablesoftware.com/ (2013)
9. JADE: Java agent development framework. http://jade.tilab.com/ (2013)
10. Kingston, J.H.: Modelling timetabling problems with STTL. In: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III. pp. 309–321. PATAT '00, Springer-Verlag, London, UK (2001), http://dl.acm.org/citation.cfm?id=646431.692901
11. Kingston, J.H.: Hierarchical timetable construction. In: Burke, E.K., Rudov, H. (eds.) Practice and Theory of Automated Timetabling VI, Lecture Notes in Computer Science, vol. 3867, pp. 294–307. Springer Berlin Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-77345-0_19
12. Kingston, J.H.: Solving the general high school timetabling problem. In: Proceedings of the 8th international conference on the practice and theory of automated timetabling (PATAT 2010). pp. 517–518 (2010)
13. Lantiv: Lantiv Scheduling Studio. http://schedulingstudio.com/ (2013)
14. Mimosa Software Ltd.: Mimosa softwre. http://www.mimosasoftware.com/ (2013)
15. Obit, J., Landa-Silva, D., Ouelhadj, D., Vun, T.K., Alfred, R.: Designing a multi-agent approach system for distributed course timetabling. In: 11th International Conference on Hybrid Intelligent Systems (HIS). pp. 103–108 (2011)
16. OpenSIS: OPENSIS. http://www.opensis.com/features.php (2013)
17. Post, G., Ahmadi, S., Daskalaki, S., Kingston, J., Kyngas, J., Nurmi, C., Ranson, D.: An XML format for benchmarks in high school timetabling. Annals of Operations Research 194(1), 385–397 (2012), http://dx.doi.org/10.1007/s10479-010-0699-9
18. Reis, L.P., Oliveira, E.: A language for specifying complete timetabling problems. In: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III. pp. 322–341. PATAT '00, Springer-Verlag, London, UK (2001), http://dl.acm.org/citation.cfm?id=646431.692907
19. Schaerf, A.: A survey of automated timetabling. Artificial Intelligence Review 13(2), 87–127 (1999), http://dx.doi.org/10.1023/A%3A1006576209967
20. Schmidt, G., Ströhlein, T.: Timetable construction an annotated bibliography. The Computer Journal 23(4), 307–316 (1980), http://comjnl.oxfordjournals.org/content/23/4/307.abstract

21. Talbi, E.G.: Metaheuristics: From Design to Implementation. Wiley Publishing (2009)
22. Time Finder: TimeFinder. http://timefinder.sourceforge.net/ (2013)
23. UniTime: University timetabling. http://www.unitime.org/ (2013)
24. Weiss, G. (ed.): Multiagent systems: a modern approach to distributed artificial intelligence. MIT Press, Cambridge, MA, USA (1999)
25. Yang, Y., Paranjape, R., Benedicenti, L.: An examination of mobile agents system evolution in the course scheduling problem. In: Canadian Conference on Electrical and Computer Engineering. vol. 2, pp. 657–660 (2004)
26. Yang, Y., Paranjape, R., Benedicenti, L.: An agent based general solution model for the course timetabling problem. In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. pp. 1430–1432. AAMAS '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1160633.1160901

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Octubre de 2015
Printing 500 / Edición 500 ejemplares