2009 Ninth International Conference on Intelligent Systems Design and Applications

Checking the Reliability of *GeSES*: Method for Detecting Symptoms of Low Performance

Javier Bravo*, Estefanía Martín†, Alvaro Ortigosa* and Rosa M. Carro*
*Department of Computer Science
Universidad Autónoma de Madrid
Escuela Politécnica Superior
C/ Tomás y Valiente 11, Madrid, Spain
{javier.bravo,alvaro.ortigosa,rosa.carro}@uam.es
†Department of Languages and Computer Systems I
Universidad Rey Juan Carlos
E.T.S. Informática
C/ Tulipán s/n, Móstoles, Madrid, Spain
estefania.martin@urjc.es

***Abstract*—In the last years the development of learning environments, and particularly of Educational Adaptive Hypermedia (EAH) systems has increased significantly. However, it is important to complement this development with evaluation methods in order to improve EAH system performance. In this context, we propose to analyze the data from student interaction with EAH systems utilizing the GeSES method. This method has been specifically designed to work with student logs and is based on C4.5 rules. In particular, the work described in this paper aims to achieve the following two objectives: testing the method with different types of data in order to find out its reliability, and detecting symptoms of low performance in a specific adaptive learning environment, called CoMoLE.**

***Keywords*-evaluation; educational; decision tree;**

I. MOTIVATION

Not all users of e-learning systems have the same goals, interests or needs. Aspects such as background, preferences, learning styles or personality may influence user-system interaction and determine the user needs. Information can be adapted to meet the user learning needs. A user model is a representation of information about an individual and is essential for adaptive systems to provide adaptation, i.e., to behave differently for different users [1]. The first applications of Adaptive Hypermedia (AH) date from the beginnings of the 90s. Brusilovsky presented the first classification of AH methods and techniques in 1996 [2]. Some e-learning adaptive systems based on these AH techniques are ELM-ART [3], AHA! [4], TANGOW [5] and CoMoLE [6].

The creation and evaluation of adaptive systems are difficult tasks for instructors and system designers. On one hand, the design process requires from the instructor to structure the knowledge and to define the mapping between this knowledge and the educational material. On the other hand, evaluation of e-Learning systems is a complex and time-consuming task, since, in most of them, only data about student interactions are available for instructors; they do not receive feedback about student behavior, learning progress, motivation, etc. Since students' behaviors are hidden inside the data recording their interactions, instructors should analyze them in order to assess student and learning environments' performance. Generally, e-Learning systems generate vast quantities of data (log files). These data consist of records of the actions done by each student while interacting with the system. In this context, one challenge is to find out a methodology for processing log files, since, normally, it is difficult to use traditional data analysis tools and techniques due to the large size of log files.

Our last works [7], [8] have demonstrated that data mining technology can be successfully used for processing log files. This is a typical application of data mining, as it blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data (discovering useful information) [9]. In this context, these previous works were centered on the evaluation of e-Learning material by exploring log files in order to find patterns indicating that some parts of the material delivered by the e-Learning environment were not adequate for students.

The *key-node* method, based on decision tree algorithms, was presented in [10]. This method detects the nodes in a decision tree that indicate symptoms of bad adaptation in an adaptive course. We also developed *ASquare (Author Assistant)*, a tool that provides a user friendly interface for evaluating adaptive courses by using data mining methods. ASquare was firstly presented in [11] and the improved version was presented in [12]. With the objective of testing the key-node method we tested it with real log files in [7]. This last work demonstrated that the application of data mining methods for evaluating e-Learning environments is useful and generated good results. However, the key-node

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.213

method obtains less accurate results for leafy decision trees. For this reason, GeSES method (based on decision rules) was developed [8]. The GeSES method proved to be useful for finding symptoms of low performance.

The goal of the current work is to generalize the proposed method presented in [8] by analyzing different data types. This paper presents the analysis of student interaction data when carrying out the learning activities included in "Operating Systems" course. The activities were delivered through an adaptive environment, which provides different levels of adaptation based on different aspects of the student model (i.e. learning styles) and also on the student context when interacting with the system.

This paper is organized in five sections. The next section describes the delivering adaptive system. The data provided for this system and the context of them are described in section three. The analysis of the data by using the GeSES method [8] is presented in section four. Finally, the last section exposes the conclusions and future work.

II. The CoMoLE system

CoMoLE (*Context-based adaptive Mobile Learning Environments*) is a mobile learning Web-based system that supports the recommendation and accomplishment of different types of learning activities (i.e. reading explanations, observing examples, test making, short open-answer exercises, collaborative problem solving, downloading electronic study material, messaging to/from other classmates, etc) [13]. Activities can be devoted to all users or to specific users. CoMoLE recommends the most suitable activities to each user according not only to his personal features and behavior but also to his context at the time of the recommendation. Adaptation capabilities and recommendation criteria are defined by means of rules. At the beginning of each session the system asks the student how much time he has available to interact with the system, and it takes this estimation into consideration when selecting the learning activities to be recommended.

CoMoLE has been used by students of "Computer Science" at the Universidad Autónoma de Madrid. A mobile learning environment was created for the course "Operating Systems", among others. It contains activities to support learning of theoretical concepts and procedures, revising examples, practicing with self-assessment tests, answering open-ended questions, reviewing already done activities and solving a problem collaboratively. Regarding adaptation in "Operating Systems", learning activities and contents were recommended to students according to their learning styles [14], activities already accomplished, results in practical tasks, and context (device used, available time and physical location at the time of the recommendation). With this aim, structural rules define the guidance to be offered to different students when accessing to groups of tasks. Furthermore, general context filters are specified (i.e. the minimum time needed to perform short_text exercises and collaborative activities for active/reflective students). Finally, specific constraints for certain activities were also included (starting or ending dates, devices to be used in collaborative activities, etc.). More details related to this environment can be found in [6].

III. Data description

The input data for this work has been provided by the Escuela Politécnica Superior at Universidad Autónoma de Madrid (Spain). Data were collected through the 2007/08 spring term from student interaction with CoMoLE, within the framework of the Operating Systems course. They correspond to 131 students, with a total of 3610 interactions with the system.

In this work, the pre-processing phase consisted of creating a large log file with data from CoMoLE log files. This task was laborious. CoMoLE generates two log files for each student: one for storing the user profile and another recording his interactions. A large log file was created by using the most useful parts of the log files for this study, and avoiding redundant parts. In addition, a cleanning phase was needed. The large log file contains information about the student profile, the student context at each time, the activities performed in each context and the outcome in each activity. It is worth mentioning that only practical activities were selected. Table I shows the composition of the student profile: a user id and values for four dimensions of learning styles, based on the Felder-Silverman model [14]:

- idUser: student identification in the system.
- Visual: visual/verbal dimension of Felder-Silverman model. This dimension is related to the manner in which contents are presented to students. Two values are available: "y" (more visual than verbal) and "n" (more verbal than visual).
- Sequential: sequential/global dimension of the model. This dimension refers to the manner in which students understand the information. Two values are available: "y" (more sequential than global) and "n" (more global than sequential).
- Active: active/reflexive dimension of Felder-Silverman model. This dimension indicates the way in which students like to process the information. Two values are available: "y" (more active than reflexive) and "n" (more reflexive than active).
- Sensitive: sensitive/intuitive dimension of the model. This dimension indicates the way in which students perceive the information. Two values are available: "y" (more sensitive than intuitive) and "n" (more intuitive than sensitive).

The next three variables for each log entry (see table II) are related to the context of the student when carrying out the activities:

- Device: type of device in which learning activities are presented to the student at this time. Two types are available in this course: PC and PDA.
- Location: place in which the student takes the activities. Three places are available: home, laboratory and others.
- timeUser: time available for the student to accomplish the activities in this session. The student sets this time when he starts a new session, and CoMoLE offers him the most suitable activities to be carried out in this period of time. Possible values for this variable are: 10 minutes, 20 minutes, 30 minutes, 1 hour, 1-2 hours, more than 2 hours.

Table III shows the variables related to the activities done, described in the following lines:

- nameAct: identification of the learning activity.
- Type: type of activity. Five types are possible: theoretical, example, self-assessment test, short free-text answer exercises (short_text) and collaborative activities.
- Finalization: it indicates if the activity is ended or not. Two values are possible: F (finished) and N (not finished).
- Score: it stores the score obtained by the student when accomplishing the activity. It is a continuous variable from 0 (the worst score) to 10 (the best one).
- Grade: this variable is not generated by CoMoLE but added artificially to each log entry. It indicates a LOW score ($0 \leq score < 5$) or HIGH score ($5 \leq score \leq 10$).

The following tables (table I, II, III) show the three parts of each log entry of three different students. The student user23 is visual (value "y" in visual), sequential (value "y" in sequential), active (value "y" in active) and sensitive (value "y" in sensitive). This student carried out the "Mem_Test1" activity at their home by using their personal computer (see table II). In addition, the student had 30 minutes to perform the activities. The next part of this log entry (see table III) indicates that Mem_Test1 is a test activity and was finished unsuccessfully (value "F" in final, value "LOW" in grade) by user23. The next log entry shows a different user with a different learning style. In this case, user73 is visual, non sequential (global), active and sensitive. User73 worked at home using their personal computer with 10 minutes available. The "MV_Test19" activity is a short_text activity and was finished unsuccessfully by the student. The last log entry shows that student user90 has the same learning style and context as user73. However, user90 finished successfully the "PagSimpleTest2" test activity.

Tables IV and V exhibit the distribution of the values of the main variables of this study within the interaction logs. For example, table IV shows that 3140 students were visual (value "y" for variable "visual"), whereas 470 where verbal (value "n" for this variable). The same information is given for the other dimensions. Table V shows that most of the

idUser	Visual	Sequential	Active	Sensitive
user23	y	y	y	y
user73	y	n	y	y
user90	y	n	y	y

Table I
EXAMPLES OF LOG ENTRIES IN THE ADAPTIVE LEARNING ENVIRONMENT (PART 1/3)

idUser	Device	Location	timeUser
user23	pc	home	30
user73	pc	home	10
user90	pc	home	10

Table II
EXAMPLES OF LOG ENTRIES IN THE ADAPTIVE LEARNING ENVIRONMENT (PART 2/3)

nameAct	Type	Final	Score	Grade
Mem_Test1	test	F	0.1	LOW
MV_Test19	short_text	F	0.1	LOW
PagSimpleTest2	test	F	10.0	HIGH

Table III
EXAMPLES OF LOG ENTRIES IN THE ADAPTIVE LEARNING ENVIRONMENT (PART 3/3)

students worked at home, since there are 3042 instances which contain the "home" value in location variable. In addition, the preferred device is personal computer (PC), with more test activities than short_text or collaborative activities. The next row in this table (grade) indicates that a great number of students failed the practical activities, since 2799 instances had a "LOW" value for this variable. It is worth noticing that the "Operating Systems" adaptive learning environment is composed of 46 different practical activities. The names of these activities are shown in the last row of the table. For instance, the first three activities are PagSimpleTest1, PagSimpleTest2 and PagSimpleTest3. These activities are listed in the table as "PagSimpleTest(1-3)".

IV. Log Analysis

The first goal of this work is to test the method GeSES [8] in order to check its reliability. In this sense, CoMoLE logs provide more information about students' profile than the log files analyzed in [8]. The second goal was to analyze CoMoLE courses by using this method to find significant patterns of behavior, which can provide support for improving the recommendations. In this context, this

Properties	Values	Number of instances
Visual	y	3140
	n	470
Sequential	y	2670
	n	940
Active	y	1364
	n	2246
Sensitive	y	3181
	n	429

Table IV
DISTRIBUTION OF MAIN DATA VARIABLES (PART 1/2)

Properties	Values	Number of instances
Device	PC	3496
	PDA	114
Location	home	3042
	lab	68
	others	500
Type	test	2306
	short_text	1235
	collaborative	69
Grade	LOW	2799
	HIGH	811
nameAct	PagSimpleTest(1-3), PagSimpleRptaLibre(1-2), SegSimpleTest(1-2), Mem_Test(1-5), PSRptaLibre(1-2), MV_Test(1-19), MultiRptaLibre(1-2), PagMV-Ejer(1-7), SegMVEjer, ColSO1, TablasRptaLibre1, TablasTest2	

Table V
DISTRIBUTION OF MAIN DATA VARIABLES (PART 2/2)

method uses C4.5 algorithm [15], which produced good results in the previous work.

The GeSES method is summarized in the following lines:

- Generating the production rules by using C4.5 algorithm.
- Selecting the rules of the ranking table (provided by C4.5 algorithm) in which the right part indicates a low score in a given activity.
- Establishing the filter limit.
- Selecting the rules that cover a number of used cases greater than the filter limit.

The first step in the analysis process was to define which attributes from the log files would be analyzed and/or used as the class variable. Considering the intention of finding activities that could present difficulties for a subset of students, eight attributes were selected: Visual, Sequential, Active, Sensitive, Device, Location, Type and nameAct. The class variable grade represents the outcome ("HIGH" or "LOW") of students when solving practical activities. In addition, the space of the problem is 26496 combinations (problem space is calculated as number of attributes and class variable combinations, therefore $space = 2{*}2{*}2{*}2{*}2{*}3{*}3{*}46{*}2 = 26496$). The problem space, together with the number of records, indicate that the data analysis is complex enough to apply this method.

After applying the C4.5 algorithm for obtaining the C4.5 rules, 43 *production rules* were generated (13 rules for class "LOW" and 30 rules for class "HIGH"). Then, the next step was selecting the rules in which the right part is "LOW". Thereby, 13 rules were selected. Table VI contains the 13 rules for class "LOW". The column size shows the number of conditions of the rule. For example, the rule 104 is $type = short\_text \rightarrow LOW[91.9\%]$ (see table VII); since the left part of this rule is composed by one attribute (*type*), its size is 1. The error column is an estimation of the number of instances classified incorrectly by the rule, i.e. the error of rule 104 is 8.10%. The accuracy of this rule is 91.9%, since $accuracy = 1 - error$. Finally, the last column points out the number of instances which cover the left side of the rule. For instance, the rule 104 is covered by 1225 instances or cases (this was the rule with the highest coverage).

idRule	Size	Error	Used Cases
104	**1**	**8.10%**	**1225**
54	**2**	**17.80%**	**418**
19	**1**	**8.50%**	**99**
30	**1**	**8.50%**	**99**
7	1	19.70%	95
26	1	9.00%	93
57	1	15.80%	81
62	1	17.10%	81
1	2	17.20%	74
37	2	18.40%	56
20	2	20.40%	54
45	3	22.30%	27
15	2	11.80%	11

Table VI
RANKING OF RULES FOR CLASS LOW

The next step was filtering the rules in order to select the most relevant. Two criteria are possible for filtering the rules: considering used cases (i.e. the cases in which all the conditions of the rule are satisfied) or considering accuracy (i.e. the percentage of correctly classified cases). Taking into account that our objective is to describe the data rather than making predictions of the data, we have chosen the former criterion. As the filter criterion is based on used cases, the table VI presents the rules ordered by this column. The filter used in [8] was 10% of the total of used cases of the rules. In this case, this filter is 241 (total number of used cases is 2413). Therefore, two rules are over this filter: rule 104 and rule 54. We prefer a filter based on third quartile or upper quartile, since more rules are selected. Then, the third quartile is 99, thus four rules are over this filter: rules 104, 54, 19 and 30. This set of rules (they are highlighted in table VI) is more representative than the set of rules provided by the first filter.

Rule 104:	type=short_text	→	LOW [91.9%]
Rule 54:	active=y, location=home	→	LOW [82.2%]
Rule 19:	nameAct=Mem_Test2	→	LOW [91.5%]
Rule 30:	nameAct=MV_Test1	→	LOW [91.5%]

Table VII
REPRESENTATIVE RULES

Rule 104 indicates that most students showed difficulties in the short_text activities. This might indicate the convenience of revising the recommendation criteria for this type of activities. The next rule demonstrates that most of active students who worked at home had difficulties. Hence, further attention should be paid to the recommendations given to active students. Rules 19 and 30 point out that most of the students had failures in activities Mem_Test2 and MV_Test1. Therefore, contents of these activities should be reviewed.

It is important to notice that the information displayed in table VII could be useful to instructors or designers of learning environments for enhancing either the recommendation criteria or the e-Learning material. It would be a valuable help to show this information to them so that they can evaluate their learning environment better.

V. CONCLUSIONS AND FUTURE WORK

This work presents the analysis of interaction data from CoMoLE log files by using the GeSES method. Two different goals were pursued in this work: testing the reliability of the method with different types of data, and detecting symptoms which could indicate problems in the recommendations offered by the system.

From this analysis we obtained four rules that could indicate the presence of this type of symptoms. One of these rules points out that most of the students presented problems when accomplishing short_text activities. The second rule shows the presence of problems for active students in most of the learning activities. These two rules show general problems: the first rule does not provide information about specific student profiles having problems with this type of activities (it seems that the score obtained is independant from the student profile); the second one does not provide information about the name or type of activities in which active students had problems (they seemed to get worse results in general, in this course). The last rules (19 and 30) contain information about particular activities in which students got bad results. Therefore, our opinion is that these four rules are important for the course responsible and should be considered to improve the course (either the recommendations or the materials provided).

It is worth pointing out that one of the main contributions of the GeSES method is to highlight relevant information that could go unnoticed when using only data mining techniques without further assistance. Moreover, this analysis showed the reliability of GeSES method, since the structure and distribution of these data were totally different from data of previous works and, even so, the method produced interesting results.

One of the features of current data mining tools is that they cannot be utilized without prior data mining knowledge. In contrast, GeSES method can be used by those without a background in data mining. In this sense, our future work will be centered on adding GeSES to *ASquare (Author Assistant)* [12] tool. In addition, further enhancements of GeSES will focus on improving the filter used within this method.

ACKNOWLEDGMENT

This work has been funded by Spanish Ministry of Science and Education through the HADA project TIN2007-64718. The authors would like to thank to Elisabeth Rodriguez who contributed in this work with her helpful comments.

REFERENCES

[1] P. Brusilovsky and E. Millan, “User Models for Adaptive Hypermedia andAdaptiveEducational Systems,” in *The Adaptive Web*, 2007, pp. 3–53.

[2] P. Brusilovsky, “Methods and techniques of adaptive hypermedia,” *User Modeling and User-Adapted Interaction*, vol. 6, no. 2-3, pp. 87–129, 1996.

[3] P. Brusilovsky, E. W. Schwarz, and G. Weber, “ELM-ART: An intelligent tutoring system on world wide web,” *Intelligent Tutoring Systems - Lecture Notes in Computer Science*, vol. 1086, pp. 261–269, 1996.

[4] P. D. Bra, A. Aerts, B. Berden, B. de Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash, “AHA! The adaptive hypermedia architecture,” in *Proceedings of 14th ACM Conference on Hypertext and Hypermedia*, New York, USA. ACM Press, 2003, pp. 81–84.

[5] R. M. Carro, E. Pulido, and P. Rodriguez, “Dynamic generation of adaptive internet based courses,” *Journal of Network and Computer Applications*, vol. 22, pp. 249–257, 1999.

[6] E. Martin and R. M. Carro, “Supporting the development of mobile adaptive learning environments: A case study,” *IEEE Transactions on Learning Technologies*, vol. 2, pp. 23–36, 2009.

[7] J. Bravo, C. Vialardi, and A. Ortigosa, “Using Decision Trees for Discovering Problems on Adaptive Courses,” in *Proceedings of E-Learn 2008: World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education*. Las Vegas, USA, 2008, pp. 268–277.

[8] J. Bravo and A. Ortigosa, “Detecting Symptoms of Low Performance Using Production Rules,” in *Proceedings of Second Educational Data Mining conference*. Universidad de Cordoba, Cordoba, Spain, 2009, pp. 31–40.

[9] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson-Addison Wesley Publishers, 2006.

[10] J. Bravo, C. Vialardi, and A. Ortigosa, “A Problem-Oriented Method for Supporting AEH Authors through Data Mining,” in *Proceedings of International Workshop on Applying Data Mining in e-Learning (ADML07) held at the Second European Conference on Technology Enhanced Learning (EC-TEL 2007)*. Crete, Greece, 2007, pp. 53–62.

[11] C. Vialardi, J. Bravo, and A. Ortigosa, “Empowering AEH Authors Using Data Mining Techniques,” in *Proceedings of 5th International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H 2007) held at the 11th International Conference on User Modeling (UM2007)*. Corfu, Greece, 2007, pp. 33–43.

[12] J. Bravo, C. Vialardi, and A. Ortigosa, “ASquare: A Powerful Evaluation Tool for Adaptive Hypermedia Course System,” in *Proceedings of Hypertext 2008 Conference*. University of Pittsburgh, Pittsburgh, USA, 2008, pp. 219–220.

[13] E. Martin, R. M. Carro, and P. Rodriguez, “A mechanism to support context-based adaptation in m-learning,” *Innovative Approaches for Learning and Knowledge Sharing - Lecture Notes in Computer Science*, vol. 4227, pp. 302–315, 2006.

[14] R. M. Felder and L. K. Silverman, “Learning styles and teaching styles in engineering education,” *Engineering Education*, vol. 78, no. 7, pp. 674–681, 1988.

[15] J. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.