A Problem-Oriented Method for Supporting AEH Authors through Data Mining

Javier Bravo[1], César Vialardi[2], and Alvaro Ortigosa[1]

[1] Escuela Politécnica Superior
Universidad Autonoma de Madrid, Madrid 28049, Spain
Email: {javier.bravo, alvaro.ortigosa}@uam.es
[2] Facultad de Ingeniería de Sistemas
Universidad de Lima, Lima 33, Perú
Email: cvialar@correo.ulima.edu.pe

Abstract. One of the main problems with Adaptive Educational Hypermedia Systems (AEHS) is that is very difficult to test whether adaptation decisions are beneficial for all the students or some of them would benefit from a different adaptation. Data mining techniques can provide support to overcome, to a certain extent, this problem. This paper proposes the use of these techniques for detecting potential problems of adaptation in AEH systems. The proposed method searches for symptoms of these problems (called anomalies) through log analysis and tries to interpret the findings. Currently, a decision tree technique is being used for the task.

1 Motivation

Whenever possible, learning systems should consider individual differences among students. Students can have different interests, goals, previous knowledge, cultural background or learning styles, among other personal features. These features should be considered in order to improve and ease the learning process for each individual. In this sense, Adaptive Educational Hypermedia (AEH) Systems [1] are able to automatically guide students, recommending them the most suitable teaching activities according to their personal features and needs. AEH systems have been successfully used in different contexts, and many on-line educational systems have been developed (e.g., AHA! [2], Interbook [3], TANGOW [4], WHURLE [5], NavEx [6] and QuizGuide [7]).

Even though AEH Systems have shown improvements over non-adaptive technology, they have not been used in real educational environments as much as its potential and effectiveness may suggest. The main obstacle to a wider adoption of AEH technology is the difficulty on creating and testing adaptive courses. One of the main problems is that teachers should analyze how adaptation is working for different student profiles. In most AEH systems, a teacher defines rather small knowledge modules and rules to relate these modules, and the system selects and organizes the material to be presented to every student

depending on the student profile. Because of this dynamic organization of educational resources, the teacher cannot look at the "big picture" of the course structure easily, as it can potentially be different for each student and many times it also depends on the actions taken by the student at runtime. In this sense, teachers would benefit from methods and tools specially designed to support development and evaluation of adaptive systems.

Due to their own nature, AEH systems collect records with the actions done by every student while interacting with the adaptive course. Log files provide good opportunities for applying web usage mining techniques with the goal of providing a better understanding on the student behavior and needs, and also how the adaptive course is fulfilling them.

With this intention, our effort is centered on helping authors to improve courses. For this reason we propose a life-cycle of an adaptive course. It is composed by *course delivering system*, *data mining tools*, *authoring tool*, and the *instructor* or *evaluator* (it can be the same person or not). The first step in this cycle is for the instructor to develop a course with an authoring tool and to load it in a course delivering system. The following step is testing the course with a group of students. Afterwards, the instructor can examine the interaction of the students with the system (log-files) with the aid of data mining tools. These tools help the instructor to detect possible failures or weak points of the course and, moreover, propose suggestions for improving the course. The instructor can follow these suggestions and make the corresponding modifications to the course through the authoring tool and load the course in the course delivering system again. Therefore, the instructor can improve the course on each cycle. However, the resulting data of applying data mining tools are pretty difficult to analyze. For this reason, it is a good idea to develop a method that helps the instructor or author to analyze data. This method is proposed in this paper. It consists of using data mining techniques and, more specifically, decision trees, to assist on the development of AEH courses, particularly on the evaluation and improvement phase. When analyzing the behavior of a number of students using an AEH system, the author does not only need to find "weak points" of the course, but also needs to consider how these potential problems are related with the student profiles. For example, finding out that 20% of the students failed a given exercise is not the same as knowing that more that 80% of the students with profile "English", "novice" failed it. In this case, the goal of our approach is not only to extract information about the percentage of students that failed the exercise but, moreover, to describe the features the students who failed it have in common.

In order to show a practical use of the method, synthetic user data are analyzed. These data are generated by Simulog [8], a tool able to simulate student behavior by generating log files according to specified profiles. It is even possible to define certain problems of the adaptation process that logs would reflect. In that way, it is possible to test this approach, showing how the method will support teachers when dealing with student data.

This paper is organized as follows: the next section describes related work in Data Mining applied to e-Learning; section three proposes a method for detecting adaptation problems in e-Learning environments; the fourth section shows two examples in which the method of the previous section is tested; and the last section exposes the conclusions and future work.

2 State of the art

Many works can be found related with e-Learning and Data Mining areas in the last years. For example, Becker and Marquardt (2004) [9] use sequence analysis with the goal of finding patterns that reveal the paths followed by the students. Merceron and Yacef (2005) [10] proposed to use decision trees to predict student marks on a formal evaluation. They also used association rules to find frequent errors while solving exercises in a course about formal logic. Pardos et al. (2006) [11] used network bayesian for predicting the score obtained of a student in an activity. Ng Cheong et al. (2006) [12] proposed to analyze interaction-logs with analysis cluster. With this analysis they determined typical errors of students in "Object Oriented Programming" subject. Romero et al. (2006) [13] proposed to use sequential patterns for recommending the next links to be shown to a student who is following an adaptive course of the AHA! system. Further information can be found in a very complete survey developed by Romero and Ventura [14]; it provides a good review of the main works (from 1995 to 2005) using data mining techniques in e-Learning environments, both for adaptive and non-adaptive systems .

3 Proposed method

AEH systems use a model of the student to adapt the material presented and the navigation support to the student features. In this way, a student is characterized by the dimensions of her student model. Attributes included on the student model are different for different AEH systems and even for different courses of the same system, and they can include, for example: previous knowledge, language, age, and learning styles, among others. If for a given adaptive course relevant attributes are, for instance, previous knowledge, language and age, the model or profile for a concrete student can contain {"advanced", "English" and "young"}.

Typically adaptive systems comprise some codification about how contents and navigation must be adapted to different student profiles. In a general way this information is coded through **adaptation rules**. According to these rules, each student can follow a different path of activities in an adaptive course, where a path is the sequence of activities visited by the student. From the teacher point of view, one of the main problems is to know if certain paths followed by the students reached successful results with more probability than others paths. In other words, it is possible that certain paths largely increase the possibilities of failure. Another problem is to know if these paths are related to a specific profile

or, on the contrary, they represent a problem not related to the adaptation but with the course in general.

A possible way of searching for problems in the adaptation rules is finding *symptoms of bad adaptation* in the user interactions with the adaptive system. In this work, we start from the assumption that problems related to the adaptation will be detected through these symptoms. Because user interactions are record on logs, a natural approach is to apply data mining, and more specifically web mining, techniques in order to find these symptoms. This approach is used in this paper for proposing a method for analyzing if the generated adapted course structure is appropriated for all student profiles. Therefore, our effort is centered on finding symptoms, inside the logs files, that indicate bad adaptation of the system. In the case study, the symptoms considered are failures in a given test. This method is described in the following lines:

- Select the entries in which the type of activity is practical activity or test. It is important that all entries must contain an indicator of success or failure of each activity. This phase is named **cleaning phase**.
- Apply the algorithm of decision trees C4.5 [15] with the following parameters:
  - Parameters: variables of student model, *name of activity* variable, and *indicator of success* variable. This indicator shows if a student pass a given practical activity or test, and two values are possible for this variable: *yes* or *no*. A value *yes* indicates that the score the student got is higher than the minimum required (and specified by the teacher). Otherwise its value is *no*.
  - Variable of classification: *indicator of success* variable.
- The resulting decision tree contains nodes for each parameters. In other words, it can be one node for each variable of student model, and one node for *name of activity* variable[3]. The leaves of the tree contain the values of the variable of classification, *indicator of success*.
- Select the leaves in which *indicator of success* variable has value **no**. In this method only these leaves are important because they indicate that many students failed a given activity.
- Analyze each path from the previous selected leaves to the root of the tree. For each path two steps are necessary:
  - Find in the path the node with the name of activity and store it. The problems in the adaptation are closely related to this activity.
  - Find in the path the values of the student profile.

The following section shows how can be applied this method.

4 Examples

In this section two examples are presented. For these examples two tools were used: **Simulog** and **Weka**[4]. Simulog, was developped in the context of this

[3] In the fig. 1 it can be shown that there are three nodes, **language**, **experience** and **age**, corresponding to the student profile, and one node **activity**, corresponding to the *name of activity*.

[4] Weka home: http://www.cs.waikato.ac.nz/ ml/weka/index.html

project [8], is a tool that simulates the log-files with symptoms of bad adaptation inside them of several student profiles. A symptom of bad adaptation is for example, most of the students with profile novice=experience fail a given practical activity. The first step in Simulog is to load the course description. Afterwards it can be specified the types of student profiles and the percentage of these profiles, the number of students to be generated, the average time that a student spent with a activity, and the symptom of bad adapation. Simulog reads the course description and, based on a randomly generated student profile, reproduces the steps that a student with this profile would take in the adaptive course. For the following examples, log files are generated for a well documented course on *traffic rules* [16]. The other tool, Weka [17], is a free software project composed by a collection of machine learning algorithms for solving real-world data mining problems. For the first example, 240 students are been simulated and for the second example 480 students are been simulated. The profiles of all simulated students are determinated by the following parameters:

- Language: Spanish (35%), English (32,5%), German (32,5%).
- Experience: novice (50%), advanced (50%).
- Age: young (50%), old (50%).

These parameters indicate that 35% of the simulated students speak Spanish and the rest 65% speak English or German. The percent of novice students is 50% and for advanced students is the same. The proportion of students that are young is 50% and for old students 50%. For example, a generated profile can be (Spanish; novice; young). In other words, students with this profile are young, speak Spanish and have novice experience.

A entry of a log file in TANGOW follows this format: <*user-id, age, language, experience, activity, complete, grade, action, activityType, activityTime, syntheticTime, success*>. Each entry belongs to an action of the student at a given point in time. Where variables user-id, age, language and experience form a student profile. The variable activity contains the activity name, complete indicates how much the student has completed the activity, variable grade stores the activity mark of the student, action is the action executed by the student (START-SESSION, FIRSTVISIT, LEAVE-COMPOSITE, LEAVE-ATOMIC)[5], activityType indicates the type of activity (Theoretical, practical), activityTime stores the time the student spent in the activity, syntheticTime stores the time when the student starts interacting with the activity, and success indicates whether the activity is considered successful or not.

4.1 Example 1

In this example we studied data on 240 students generated by Simulog, corresponding to following symptom of bad adaptation: 70% of students with profile **language="Spanish", experience="novice", age="young" fail** the **S_Ag_Exer** activity.

[5] For this work only is important LEAVE-ATOMIC meaning a student leave an atomic activity (more details in [18]).

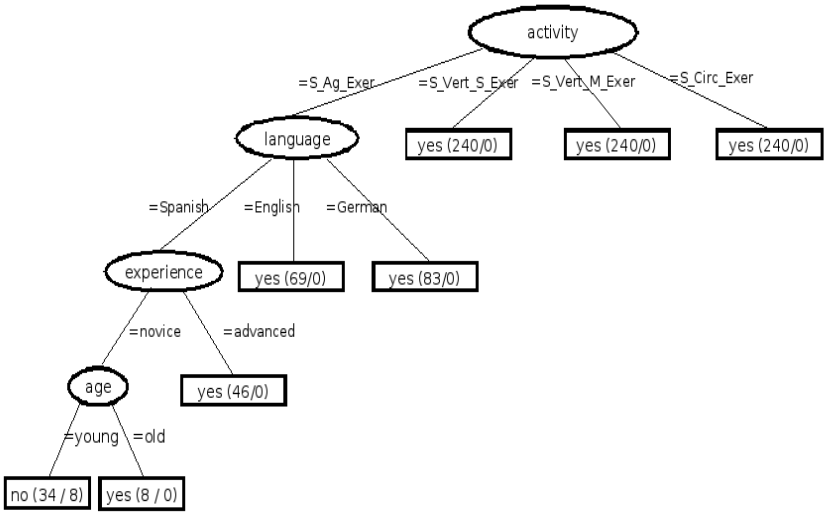

Fig. 1. Decision tree in the example 1

According to the previous method, the first step (cleanning phase) was to clean the data. It consists of removing from logs the records that are not necessary for the mining phase. Cleaning in this case, is both important and necessary for the size of data as a whole, and consequently, for the speed and accuracy with which results are obtained. With this intention, the records with action different of LEAVE-ATOMIC were eliminated. Afterwards the records with type of activity different of "P" (test or exercise activities) were eliminated also. Therefore, the final set of records for analizing contained 960 records. This task is adequate for all data mining processes that contain data that do not supply information for pattern construction. The second step is to generate the decision tree (j48 with 0.25[6] of confidence factor). The figure 1 shows the obtained decision tree. The last step is to find the node activity and the profile, and it is described as follows:

– Only it is found in the tree one leaf with the value no. This leaf has 77% of well classified instance, and this proportion is significant. The value of node activity for this leaf is again S_Ag_Exer. The profile is formed by age="young", experience="novice", language="Spanish".

Therefore, this tree indicates that a great number of the students who speak Spanish, who have novice experience and who are young had many failures in the S_Ag_Exer activity. It is important to highlight that in this example the tree has a high percentage of well classified instances. This fact is due to absence of randomness effect in variable grade when a student is not related to the symptom of bad adaptation. In this case, these students always pass the activity.

[6] This confidence factor is a good value for prunning and for avoiding the overfitting.

4.2 Example 2

In this last example data from 480 students were studied, generated by Simulog with two symptoms of bad adaptation and randomness effect in the variable grade. Therefore, in this example there are two sources of noise, the number of symptoms and the randomness effect. These symptoms were defined as 60% of students with profile (Spanish; novice; young) fail the S_Ag_Exer, and 60% of students with profile (English; novice; young) fail the S_Circ_Exer activity. The first phase is to proceed to clean the data (cleanning phase) as in the other example. The results showed 1920 records to which the algorithm of decision tree (j48 with confidence factor of 0.25) is applied in the second step (see figure 2). The last step of the method obtained the following outcomes:

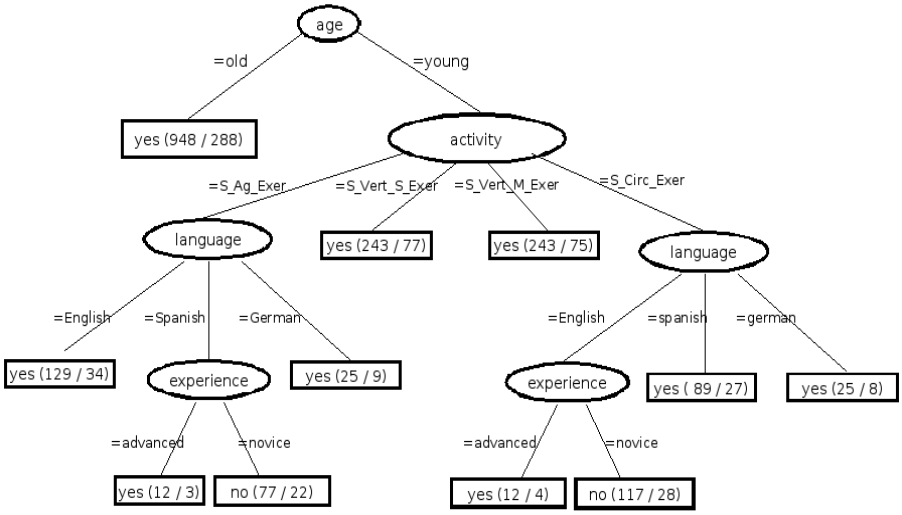

Fig. 2. Decision tree in the example 2

- Two leaves with the value no are found in the tree. Two activities are related to these leaves: S_Ag_Exer and S_Circ_Exer. Therefore two possible anomalies can be found.
- For the first leaf no (related to the node activity=S_Ag_Exer) the student profile is defined by the variables experience="novice", language="Spanish" and age="young".
- For the second leaf with no value (related to the node activity=S_Circ_Exer) the student profile is defined by the variables experience="novice", language="English" and age="young".

Thus, two symptoms of bad adaptation are detected, since the proportion of well classified instances is reasonably high in both leaves with **no** value (more

than 70%). Hence, the young students with novice experience who speak Spanish had many difficulties with S_Ag_Exer activity. Besides, there was another group of young students with novice experience with many difficulties in S_Circ_Exer activity, but the language in this group was English.

5 Conclusions

This work proposes a practical way, based on decision trees, to search for possible wrong adaptation decisions on AEH systems. The decision tree technique is a useful method for detecting patterns related to symptoms of potential problems on the adaptation procedure.

This paper presents two experiments intended to show the advantages of this method. They were carried out with different number of simulated students and also with different percentages of students failing the same exercise, all of them corresponding to a certain profile. The first experiment proves the effectiveness of decision trees for detecting existing symptoms of bad adaptation. The second experiment was carried out with a larger amount of students. Moreover, noise was included in the data through a randomness factor in the grade variable. It was added with the objective of generating data to be closer to reality. This experiment shows the algorithm scalability and reliability. Furthermore, the method for detecting symptoms provides instructors with two types of information. On one hand, the instructor can know whether a symptom is closely related to a given activity. Then, she can decide to check the activity and the adaptation around it. On the other hand, the instructor can detect whether a group of students belonging to a certain user profile (or sharing certain features) has trouble with an activity. Then, she can decide either to modify the activity itself, to include additional activities to reinforce the corresponding learning, to establish previous requirements to tackle the activity or to change the course structure, i.e., for students matching this learning profile, by incorporating rules to represent the corresponding adaptation for this type of students.

The usefulness of this method for detecting potential problems in adaptive courses has been shown. However, to be useful for instructors this method ought to be supported by tools which hide the technique details to non expert users in data mining. In that sense, we are working for adding this method in **ASquare**[18].

The utility of decision trees for this work is not centered on the accuracy when predicting the success of students when tackling learning activities. Therefore, the percentage of well classified events is less important than the capability of this tree to show the symptoms of bad adaptation.

Finally, the examples presented in this work show that, although decision trees are a powerful technique, they also have weak points. An important weakness is that the information extracted may not always be complete, since algorithm C4.5 works with probabilities of events. Therefore, for complementing the information extracted it may be necessary to use this method together with other data mining techniques such as association rules, clustering, or other mul-

tivariable statistical techniques. In that sense, our future work is centered on testing the combination of decision trees with other techniques for completing the information extracted from those. Other important challenge is to know the threshold index of failures that indicates a symptom of bad adaptation.

Acknowledgment

This work has been partially funded by the Spanish Ministry of Science and Education through project TIN2004-03140 and TSI2006-12085. The author C. Vialardi is also funded by Fundación Carolina.

References

1. P. Brusilovsky. Developing adaptive educational hypermedia systems: From design models to authoring tools. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environment*, pages 377–409. Dordrecht: Kluwer Academic Publishers, 2003.
2. P. De 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*, pages 81–84. Nottingham, UK, 2003.
3. P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: A tool for developing adaptive courseware. In *Proceedings of 7th Intl. World Wide Web Conference*, volume 30, pages 291–300. Brisbane, Australia, 1998.
4. R.M. Carro, E. Pulido, and P. Rodriguez. Dynamic generation of adaptive Internet-based courses. *Journal of Network and Computer Applications*, 22:249–257, 1999.
5. A. Moore, T.J. Brailsford, and C.D. Stewart. Personally tailored teaching in WHURLE using conditional transclusion. In *Proceedings of the Twelfth ACM conference on Hypertext and Hypermedia*. Denmark, 2001.
6. M. Yudelson and P. Brusilovsky. NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples. In C. K. Looi, G. McCalla, B. Bredeweg, and J. Breuker, editors, *Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED)*, pages 710–717. Amsterdam, The Netherlands, IOS Press, July 2005.
7. S. Sosnovsky and P. Brusilovsky. Layered Evaluation of Topic-Based Adaptation to Student Knowledge. In *Proceedings of Fourth Workshop on the Evaluation of Adaptive Systems at 10th International User Modeling Conference*, pages 47–56, July 2005.
8. J. Bravo and A. Ortigosa. Validating the Evaluation of Adaptive Systems by User Profile Simulation. In Stephan Weibelzahl and Alexandra Cristea, editors, *Proceedings of Workshop held at the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006)*, pages 479–483. National College of Irland, Dublin, Ireland, June 2006.
9. K. Becker, C.G. Marquardt, and D.D. Ruiz. A Pre-Processing Tool for Web Usage Mining in the Distance Education Domain. In *Proceedings of the international Database Engineering and Application Symposium (IDEAS04) 2004 IEEE*, pages 78–87, 2004.

10. A. Merceron and K. Yacef. Educational Data Mining: a Case Study. In C. Looi; G. McCalla; B. Bredeweg; J. Breuker, editor, *Proceedings of the 12th international Conference on Artificial Intelligence in Education AIED*, pages 467–474. Amsterdam, The Netherlands, IOS Press, 2005.
11. Z.A. Pardos, N.T. Heffernan, B. Anderson, and C.L. Heffernan. Using Fine-Grained Skill Models to Fit Student Performance with Bayesian Networks. In *Proceedings of Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*, pages 5–12. Jhongli, Taiwan, June 2006.
12. M-H. Ng Cheong Vee, B. Meyer, and K.L. Mannock. Understanding novice errors and error paths in Object-oriented programming through log analysis. In *Proceedings of Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*, pages 13–20. Jhongli, Taiwan, June 2006.
13. C. Romero Morales, A. R. Porras Pérez, S. Ventura Soto, C. Hervás Martínez, and A. Zafra. Using sequential pattern mining for links recommendation in adaptive hypermedia educational systems. *Current Developments in Technology-Assisted Education*, 2:1016–1020, 2006.
14. C. Romero and S. Ventura. Educational Data Mining: a Survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146, 2007.
15. Tom Mitchell. *Decision Tree Learning*, chapter 3, pages 52–73. McGraw Hill, 1997.
16. R.M. Carro, E. Pulido, and P. Rodriguez. An adaptive driving course based on HTML dynamic generation. In *Proceedings of the World Conference on the WWW and Internet WebNet99*, volume 1, pages 171–176, October 1999.
17. I.H. Witten and E. Frank. *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2005.
18. C. Vialardi, J. Bravo, and A. Ortigosa. Empowering AEH Authors Using Data Mining Techniques. In *Proceedings of Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H) held at the 11th International Conference on User Modeling (UM2007)*. Corfu, Greece, June 2007.