Journal of Universal Computer Science, vol. 14, no. 17 (2008), 2777-2798
submitted: 30/6/08, accepted: 27/9/08, appeared: 28/9/08 © J.UCS

Improving AEH Courses through Log Analysis

César Vialardi
(Universidad de Lima, Peru
cvialar@correo.ulima.edu.pe)

Javier Bravo, Alvaro Ortigosa
(Universidad Autónoma de Madrid, Spain
http://tangow.ii.uam.es/opah
{Javier.Bravo, Alvaro.Ortigosa}@uam.es)

Abstract: Authoring in adaptive educational hypermedia environment is complex activity. In order to promote a wider application of this technology, the teachers and course designers need specific methods and tools for supporting their work. In that sense, data mining is a promising technology. In fact, data mining techniques have already been used in E-learning systems, but most of the times their application is oriented to provide better support to students; little work has been done for assisting adaptive hypermedia authors through data mining. In this paper we present a proposal for using data mining for improving an adaptive hypermedia system. A tool implementing the proposed approach is also presented, along with examples of how data mining technology can assist teachers.

Keywords: authoring support, adaptive educational hypermedia, data mining applications
Categories: I.2.6, J.7, K.3

1 Introduction

Adaptive Educational Hypermedia (AEH) Systems [Brusilovsky, 03] automatically guide and recommend teaching activities to every student according to their needs, with the objective of improving and easing his/her learning process. AEH systems have been successfully used in different contexts such as in the classroom and the home. In addition, many on-line educational systems have been developed (e.g., AHA! [De Bra, 03], Interbook [Brusilovsky, 98], TANGOW [Carro, 99] and WHURLE [Moore, 01]). These systems adapt their educational contents to different dimensions of the learner profile, such as: current knowledge level, goals educational context (e.g., if they are at the school, university, or learning from home), or learning styles [Cassidy, 03] [Paredes, 04], among others.

As it was described by Brusilovky [Brusilovsky, 03], the secret of adaptivity in adaptive hypermedia systems is the "knowledge behind the pages". AEH systems model the domain knowledge in a manner most conductive to the students. In order to let the adaptive system know what to present at a given moment to a particular student, the author of the AEH system needs to structure the knowledge space and to define the mapping between this space and the educational material. Moreover, this mapping should be different depending upon student profiles in order to facilitate learning to each of them according to their specific needs. This individualizing aspect of making student profiles further complicates the AEH design.

As a result, a lot of effort is required to design AEH systems and even more effort is needed to test these systems. The main problem is that teachers should analyze how adaptation is working for different student profiles. In most AEH systems, the teacher defines rather small knowledge modules and rules to relate these modules, and the system structures the material to be presented to every student on the fly, depending on the student profile. Due to this dynamic organization of educational resources, the teacher cannot look at the “big picture” of the course structure, because it can potentially be different for each student. Furthermore the evaluation of the course is made more difficult due to the lack of feedback that teachers usually have in traditional classrooms. Even if results of tests taken by distance learners are available, they can provide hints about what the student knows or not, however they provide little information on the material presented to this particular student or about his/her interaction with the system.

In order to reach a wider adoption of AEH systems, the teacher’s work should be made easier, through methods and tools specially designed to support development and evaluation of adaptive material. In this context, we propose the use of data mining techniques to assist on the authoring process.

AEH systems, as any web based system in general, are able to collect a great amount of user data on log files, that is, records with the actions done by the user while interacting with the, i.e. adaptive course. Web usage mining tries to discover usage patterns in these log files through data mining techniques. The goal is to understand and better serve the user and the application itself [Srivastava, 00a].

These techniques are being used by many organizations that provide information through web based environments, such as e-Commerce and E-learning devoted organizations [Zaïane, 01]. On the e-Commerce context, data mining is being used as an intelligent tool developed with the goal of understanding online buyers and their behavior. Organizations using these techniques have experienced increases on sales and benefits. To reach this goal, data mining applications use the information that comes from the users to offer them what they really need [Srivastava, 00b]. Ultimately the goal of data mining in e-Commerce is to enhance the customer experience while retaining and enhancing site/brand loyalty.

In the E-learning context the objective is two-fold. On the one hand, data mining is used to analyze student behavior and provide personalized views of the learning material, which otherwise is the same for all the students [Romero, 05]. On the other hand, data mining seeks to fulfill the needs of the instructors, that is, to provide a proper vision of the education resources. The ultimate objective is to find out if students are learning in the best possible way, which is, of course, a goal very difficult to quantify and qualify.

From the teacher’s point of view, web mining has the objective of mining the data about distance education in a collective way, just as a teacher would do in a classroom when he/she adapts the course for a student or a group of them [Zaïane, 02]. The data mining takes care of finding new patterns from a large number of data.

In this work we demonstrate how data mining techniques can be used to discover and present relevant pedagogic knowledge to the teachers. Based on these techniques, a tool for teachers was developed to support the evaluation of adaptive courses. In order to demonstrate the practical use of the tool and methods, synthetic user data was analyzed. These data are generated by Simulog [Bravo, 06], a tool able to simulate student behavior by generating log files according to specified profiles. It can also define certain problems (of the adaptive material) that logs would reflect. In that way, it is possible to test the evaluation tool, showing how the tool will support teachers when dealing with student data.

The next section briefly describes the state of the art and related work. Section 3 shows the architecture of the system, as well as already existing tools, upon which the approach was implemented. Section 4 explains how to use data mining for supporting AEH authors and section 5 describes a tool built based on those techniques. Finally, section 6 outlines the conclusions.

2 State of the Art

The internet has quickly transformed the way the people communicate with each other, do business, think, etc. Furthermore, it becomes one of the most fundamentally integrated platforms for information distribution. It produced an increase of attempts to customize the information delivery. As a result, AEH systems provide more user friendly learning applications, such as independent platforms and free locations. However, an outstanding challenge is to collect and classify specific data of users (knowledge levels, types of cultures and motivations) in order to adapt the information delivery.

In the last few years a number of works have focused on using data mining techniques in E-learning systems. The most widespread techniques are classification algorithms and association rules. Classification algorithms search common properties between registry sets in different classes according to a classification model [Chen, 96]. These techniques are most suited for predicting or describing data sets with binary or nominal categories, but they are less useful for ordinal categories because they do not consider the implicit order between categories. Regarding classification techniques, decision trees are used in this work. Association rules [Agrawal, 93] are relevant relationships found in data items. The use of rule mining in education is not new and has already been successfully employed in several AEH systems. One of the first works based on association rules in AEH domains was proposed by Zaïane [Zaïane, 06]. This work focuses its investigation on two basic points: the first point is to give automated support to students who take an online course proposing the use of advising systems; the second, is to support the instructor in identifying student behavior patterns, based on the information that students leave when taking online courses. Zaïane proposes suggesting activities that could be beneficial to ensure the E-learning through shortcuts or jumps in order to help students improve browsing. In this browsing of the course material using the best path as this way to facilitate the fulfillment of their objectives. This scientific research looks for patterns of course use in the AEH environment in order to validate them or to improve them. Hence, it uses the evaluations obtained in the practical activities taken by each student with the objective of determining the student's relation with the student's profile. For this reason two techniques are used: decision trees and association rules, and like in the case of Zaïane, the objective is to support the instructor so that by means of the obtained knowledge, course content and structure can be improved.

García and Romero [García, 07], run a comprehensive study on the advantages and disadvantages of using association rules for AEH courses. This work details a series of steps to follow and obtain association rules. Collecting data, data pre-processing, applying the mining algorithms and data post-processing are the four steps that also are used in this study to obtain the association rules described in section 5.3.

Merceron and Yacef published another study where association rules are extracted from data of Logic-ITA [Merceron, 05]. Logic-ITA is a Web-based tutoring tool used to offer courses, specifically to help students in their formal logic exercises. This system can also report the progress of students to their teacher. Similar to our study, Merceron and Yacef focused their work on detecting patterns of mistakes made by the students by using association rules. They found a relation between a given number of mistakes of students applying relevance methods such as Chi square, cosine and contrasting rules [Merceron, 07]. In this study, the association rules are used to relate the categories of the student profile to the success or failure in practical exercises; nevertheless, Merceron uses the association rules to find errors as students solve exercises, displaying results in real time.

In a different research study, Superby used data mining and statistical techniques to determine factors that influenced success in first year college students [Superby, 06]. The goal of this study was to classify students in three different groups: low, medium and high risk of failure in university level study. This study tries to explain the different academic performances of students by predicting the probability of success of any given student.

Similar to our study, Superby uses ID3 algorithm [Quinlan, 93], obtaining as in our case, a tree that has the advantage of being relatively simple to interpret.

Becker and Marquardt's study [Becker 04] showcases the use of sequence analysis, where student logs are analyzed in order to find patterns that reveal the different paths taken by the students. They used this information to improve the student experience. This case focuses mainly on two problems that exist in E-learning environments. First, the behavior of the student in terms of the navigation structure expected to be followed; and second, the use of resources available to carry out learning activities. The first of these problems was the bases for this research, since our objective is to apply data mining techniques to log files. Log files are patterns that describe student behavior in the practical activities of a course. The association between the success and failure is related to the student's profile. Our work presented in this paper differs from the authors mentioned above through our attention on student profiles. While Becker looks for navigation patterns without considering student profiles, our work is centered on finding behavioral patterns in the outcomes (success or failure) of student practical activities in AEH courses.

3 Antecedents and Existing Tools

Our goal is to analyze interaction data looking for opportunities of improvement on the adaptive course design. That is, we propose to use data mining techniques for finding patterns in the data and interpreting them, in order to present the AEH designer recommendations on how the course structure and contents can be enhanced.

Even if the approach is independent of any particular AEH system, we have designed a concrete architecture for applying the ideas within the context of an existing AEH system. In this way, figure 1 shows the context of use for the data mining techniques and the evaluation tool proposed on this work. Basically, the teacher or course designer uses some of the authoring tools available for creating an adaptive course. Afterwards, students take the course, which is delivered by TANGOW (Task-based Adaptive learNer Guidance On the Web) [Carro, 99] AEH system, which adapts both the course structure and the contents presented according to each student features. Finally the teacher, using the Author Assistant, can analyze the student performance and eventually, he/she will be able to improve the course with the information obtained. Besides, the evaluation tool itself can be tested by using synthetic logs generated by *Simulog* [Bravo, 06]. The main components of this architecture are explained in the following subsections (Simulog is explained in section 6).

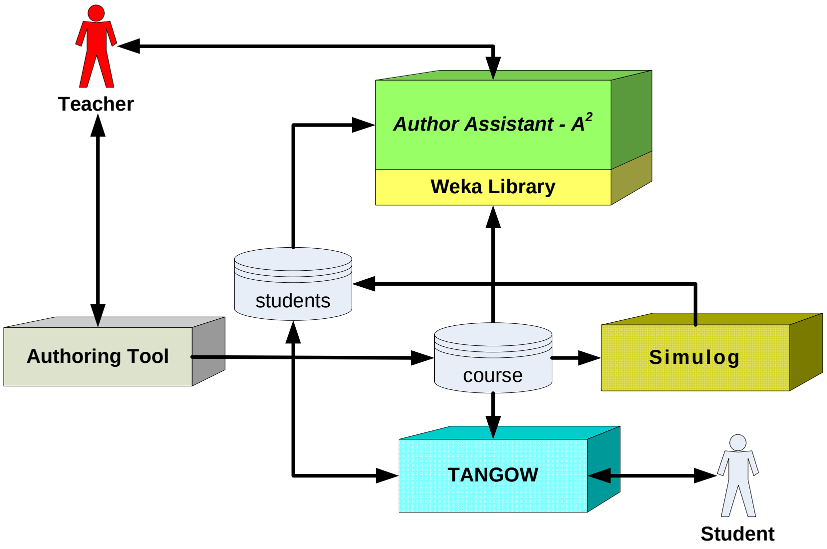


Figure 1: General architecture of the system where the evaluation tool (A^2) has been included.

3.1 TANGOW

AEH systems adapt contents and materials presented throughout the course and customize the navigation according to each student profile (or student model). The student profile stores the attributes relevant for adaptation, which can be different for different AEH systems or even for different adaptive courses on the same AEH system. For example, a student profile in an adaptive course developed in TANGOW can be specified by {experience = “advanced”; language = “English”; age = “young”}. These three values indicate that the student takes an English course version, he/she has advanced value in *previous knowledge* attribute, and his/her *age* is under 18 years old.

Commonly, an adaptive system includes some type of coding about how contents and navigation must be adapted for different student profiles. In general, this information is coded through adaptation rules. According to these rules, each student may continue different activities' paths in an adaptive course, where a path is the sequence of activities visited by the student. A TANGOW-based course is formed by *teaching activities* and *rules*. A teaching activity is the basic unit of the learning process. The main attributes of a teaching activity are **type,** which can be *theoretical* (T), *practical* (P) or *example* (E); and **composition type**, which can be either *atomic* (A) or *composite* (C), this is to say, an activity is composed by other (sub)activities.

The way in which activities are related to each other is specified by means of rules. A rule describes how a composite activity is built up of sub-activities, sets the sequencing between them, and can include preconditions for its activation, related either to user features or to requirements regarding other activities. Triggering a rule determines the next activity or sub activities that will be available to a given student, based on his/her profile [Carro, 99a**]**. A student profile is composed of pairs attribute name – value, usually called dimensions. The dimensions relevant for a given course are defined by the teacher and can be used on the conditions needed for triggering a rule.

The examples of adaptive courses and activities shown through this paper correspond to an adaptive course on *traffic rules*, well documented in previous works [Carro, 99a].

3.2 Weka

Weka [Witten, 05] is a free software project composed by a collection of machine learning algorithms for solving real-world data mining problems. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. The tools can be accessed through a graphical user interface and through a JAVA API; this last method is used in this work.

4 Mining TANGOW logs

Data collection is vital for the data mining process. Without adequate data it is very unlikely to extract any useful information to understand the student behavior and to extract conclusions about the course. In the next subsections the data produced by TANGOW/Simulog are described, as well as the data preparation process applied. The two data mining techniques used on the tool, *Classification* and *Association Rules*, are also presented.

4.1 Data Preparation

In the data mining context, this phase is named pre-processing. It consists of selecting and transforming the data from different sources, making it ready to be analyzed. Cooley [Cooley, 99] proposes a step by step data preparation technique for web mining. These steps can be applied to adaptive hypermedia environments and also to

E-learning. However, the decision about whether to use each phase or not depends on the starting point and the final goal.

Little pre-processing was needed on this work because the data was specifically generated for the data mining analysis. That is to say, the logs generated by TANGOW were tuned to fit the requirements of the data mining tool, and only two of the data-preparation phases were used.

- Data cleaning: this task is responsible for removing from the log files the registers that are not necessary for the mining phase. Cleaning these files is important for the precision of the results. In TANGOW/Simulog, interaction data of each student is generated in a different file. The cleaning module will also be required to parse the data and gather all of them into one big file.
- Data filtering: the goal is to extract a relevant data subgroup, depending on the specific data mining task to be done. In this initial phase, it was also decided to begin considering only *practical* activities, since TANGOW is only able to assess the progress of the students in the practical activities.

Attributes	**Description**
Activity	Activity id
Complete	It represents the level of completeness of the activity. If the activity is composed, it takes into consideration the completeness of all sub-activities. It is a numeric parameter that ranges from 0 to 1. Value 0 indicates the activity was not completed and value 1 indicates the activity was fully completed
Grade	The grade given to each activity. In practical (P) activities it is usually calculated from a formula provided by the teacher. In composed activities it is the arithmetic mean of sub-activity grades. Value 1 indicates the activity was finished with success and value 0 indicates the activity was finished with failures.
NumVisit	Number of times the student has visited the activity.
Action	The action executed by the student; these are defined by the TANGOW system: "START-SESSION”: beginning of the learning session. "FIRSTVISIT": first time an activity is visited. "REVISIT": any visit to the activity following the first one. "LEAVE-COMPOSITE": the student leaves the (composed) activity. “LEAVE-ATOMIC": the student leaves the (atomic) activity.
ActivityType	The type of activity: *theoretical* (T), *exercises* (P) and *examples* (E).

Table 1: Description of parameters of interaction data.

4.2 Data Description

TANGOW generates one log file for every student and course where he/she is enrolled. The first section of a log file contains a description of the student profile, that is, a list of attributes names and values used on the adaptation rules. The second section is composed by a list of entries, each one containing the data described on [Table 1].

An example of the interactions of the student *s1* with the system is shown in [Table 2]. This table shows the student starts the course named *EduVial* (main composed activity of the course). After two minutes he/she leaves (action LEAVE-COMPOSITE) this activity and he/she starts (action FIRSTVISIT) the activity *Req_conductor* that is a theoretical activity (activityType=T). Parameters *complete* and *grade* have values 0.0 because it is the first time the student visits the activity named *Req_conductor*.

<entry activity="EduVial" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T12:58:10" type="FIRSTVISIT" />
<entry activity="EduVial" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T13:00:27" type="LEAVE-COMPOSITE" />
<entry activity="Req_conductor" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T13:00:27" type="FIRSTVISIT" />
<entry activity="Req_conductor" activityType="T" complete="1.0" grade="1.0" numvisits="1" timestamp="2007-05-10T13:18:04" type="LEAVE-ATOMIC" />
<entry activity="S_Signs" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T13:18:04" type="FIRSTVISIT" />
<entry activity="S_Signs" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T13:27:06" type="LEAVE-COMPOSITE" />
<entry activity="S_Types" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T13:27:06" type="FIRSTVISIT" />
<entry activity="S_Types" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T13:48:17" type="LEAVE-COMPOSITE" />
<entry activity="S_Ag" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T13:48:17" type="FIRSTVISIT" />
<entry activity="S_Ag" activityType="T" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T14:04:09" type="LEAVE-COMPOSITE" />
<entry activity="S_Ag_Exer" activityType="P" complete="0.0" grade="0.0" numvisits="1" timestamp="2007-05-10T14:04:09" type="FIRSTVISIT" />
<entry activity="S_Ag_Exer" activityType="P" complete="1.0" grade="1.0" numvisits="1" timestamp="2007-05-10T14:26:50" type="LEAVE-ATOMIC" />

Table 2: Interactions of a given student.

Afterwards, he/she visited the activity named *S_Ag* (*signs of traffic policeman theory*) and started the first practical activity called *S_Ag_Exer (exercises of signs of* *traffic policeman).* The student needed for finishing this activity 22 minutes, and he/she completed it with success (complete=1 and grade=1).

4.3 Data Mining

Data mining consists of analyzing certain information and applying appropriate algorithms in order to extract patterns from the original data. The challenge of data mining is precisely to work with huge amounts of data from information systems. Moreover, these data could contain their particular problems: noise, incomplete data, volatility, etc.

Data mining is a multidisciplinary field that involves techniques such as automatic learning, pattern recognition, statistics, and databases. It is considered as a natural evolution initiated with the creation of the first databases. What data mining intends to do is to automate the process by localizing and extracting hidden patterns. In its purest form, Data Mining does not look for a specific type of information, but simply searches patterns that exist in the data.

Data mining tasks are divided in two big categories. The first one consists of predictive tasks, whose objective is to predict the value of a particular attribute, based on values of other attributes. The attribute to be predicted is commonly known as dependent variable, while the attributes used to make predictions are known as explicative or independent variables. The second category of tasks includes descriptive activities whose objective is to produce patterns (correlations, cluster tendencies, trajectories and symptoms) that synthesize the underlying relationships in the data.

One of the most used predictive techniques involves decision trees. They are diagrams constructed from a set of observations from some domain, and they have attributes describing each element. Each diagram represents and classifies a series of conditions concerning the values of the attributes. The objective is to obtain a conclusion, called objective function, about the observations.

In its most basic form, a decision tree is composed by nodes, leaves and edges. A node is associated to attributes that are evaluated to determine the path to be followed through two or more possible alternatives (represented by edges). The first node evaluated is known as root node. However, a node that does not branch itself is known as leaf, and is a result that the tree (objective function) gives back. Finally, a path that starts at the root node and finishes in a leaf, is named branch, and represents those instances that fulfill the conditions of the nodes evaluated in the path. In other words the decision trees classify instances. Therefore, they can also be used as a generalization of those instances that have not been included in the training examples. This is to say, a decision tree is a predictive model.

According to [Mitchell, 97], decision trees represent a “disjunction of conjunctions of restrictions” of the values of the attributes of the instances. Consequently, each path corresponds to the conjunction of values to which the attribute has been subordinated. Therefore the tree itself is the disjunction of these conjunctions.

One of the basic algorithms used to construct a decision tree is called ID3 (Inductive Decision Tree) [Quinlan, 93]. It uses an up-bottom search through the whole space of possible decision trees. The ID3 algorithm constructs the decision tree until it perfectly classifies the training examples, or until every attribute has been used. The entry data for the ID3 algorithm are known as the sets of “training” or "learning" instances that the algorithm will use to generate the decision tree. The ID3 algorithm constructs the decision tree evaluating each instance to determine if it classifies correctly the training examples. As a result, this attribute decides the attribute to be analyzed. The best attribute is chosen to be the root node. Then, an edge is extracted from each possible value of the attribute. The training examples are classified according to the existing alternatives. New nodes are created for each one, repeating the same procedure; i.e., always choosing the best attribute.

Association rules are found within the descriptive tasks. These pursue the objective of obtaining a pattern, in the form of rules. This pattern is capable of describing the relationships among the different attributes of a data base. Besides, these patterns represent patterns of behavior based on the joint appearance of values of two or more attributes. The association rules are one of the simplest and useful ways of producing knowledge [Agrawal, 93]. Initially they were used to discover relationships among items in the "market basket" transaction analysis. They discovered rules such as: "X% of clients that buy item A also buy item B" and also "a person that buys a set of X items tends to buy a set of Y items". Different from other classification methods, in the second part of the statement, there can be more than one value corresponding to some attribute.

In the context of mining web uses, the association rules can be used to discover frequently followed paths by a group of visitors to some website. Later this information can be used to restructure the mentioned website.

Applications of association rules to adaptive hypermedia environments allow us to discover associations among different elements according to the situation. For example, associations or relationships can exist among learned concepts, learning sessions, student done activities, student actions, time spent by students in completing the activities regarding their performance, etc. Generally two measures are used to know the quality of the rule: *support* and *confidence*. The *support* of a rule is defined as the number of instances that fulfill the antecedent correctly. On the other hand *confidence* measures the percentage of times the rule predicts correctly with respect to the number of times it is applied. For example, a high level of confidence represents a precise rule, while a low support represents low frequency case that probably does not reveal representative information of the transactions.

Association rules are obtained applying the "A-priori" algorithm, which is the algorithm for association rule learning. The goal of this algorithm is to obtain a set of rules from the supplied data and to select those rules with the most information content. The algorithm searches sets of items that are over a minimum coverage level defined by the analyst. The procedure is started with this premise building sets made by a single item. Later on, this set is used to build the two item set, and so on. The algorithm finishes when it reaches a size in which the set of selected items is empty [Agrawal, 93].

5 The Author Assistant Tool (A^2)

Based on the *Weka* collection of data mining algorithms, a tool able to assist the evaluation of adaptive courses was built. The tool, named **Author Assistant** (**A^2**), will be presented through a use case where two different algorithms will be applied. **A^2** is used to analyze the logs generated from the student interactions with the system, and it can provide initial hints about potential problems on the course, an even suggest actions oriented to solve the problems. In this sense, **Author Assistant** is particularly valuable when the adaptive course is offered periodically or continuously to the students. The data analyzed in the examples are synthetic logs generated by *Simulog*. In that way, it is simpler to know what kind of problem **A^2** should find and provide easier-to-understand data. However, the final goal is to use **A^2** with real student logs.

5.1 Use Case

This case has been denominated "warning notification" since it will help to show how data mining on A^2 can trigger alerts of potential problems, so that teachers can improve their courses. It is important to note that this case study is longitudinal. In other words, it analyses the different visits of a student for each activity at different times. Each student is represented repeated times in the log file for each activity. However, results and their respective interpretations must be observed carefully. None of the conclusions should be interpreted with respect to any particular student because it is not the student who is analyzed in each case, but it is analyzed the tendency of the group while it performs a practical activity, including the number of times the student visits it after a withdrawal or a failure.

Description: The data simulate the interaction of a group of 100 students taking the *traffic* course in the TANGOW system. Once the synthetic data is generated, the processing of the data will be done using the *Weka* data mining tool.

Objective: The fundamental objective of this experiment is to find, by means of data mining processes, additional knowledge that can throw an alert signal to the teacher to help his/her to:

- Modify a critical part of the course when he/she realizes that most of the students of a specific profile fail on a certain exercise.
- Identify groups of students with problems of performance on the course.

Synthetic Data Generation: Data generation was done according to the dimensions shown in [Table 3]. In other words, students with different previous experience will be generated (*novice* and *advanced*), with four different languages (*English*, *Spanish*, *German and Italian*) and with two different ages (*young* and *old*). It is important to note that the attribute *language* indicates the language of the contents of the adaptive course. The generated interaction data contain the following anomaly or symptom of bad adaptation:

• Profile	: Experience = novice, Language = any, Age = any
• Activity	: *S_Ag_Exer*
• Type of anomaly	: Abandon
• Proportion	: 70%

This anomaly represents that 70% of the students with "experience = novice" abandon the activity *"exercises of signs of traffic policeman"*. In other words, a *novice student* abandons the *S_Ag_Exer* with 70% probability. In this case the *abandoning* property indicates that a student leaves the exercise before completing it. In this case 12,800 registers were generated. However, the amount of data is reduced to 800 registers after the cleaning phase. This phase consists of removing the registers from the log file that are not necessary for the mining phase. Therefore, all registers with a *type* different from LEAVE-ATOMIC were eliminated. Afterwards, all registers with an *activity type* different from practice activities (P) were eliminated from the remaining registers.

Dimension	**Values**
Student experience	Novice, Advanced
Student languages	English, Spanish, German, Italian
Student age	Young, Old

Table 3: Student profiles (dimensions considered for the course)

A typical entry in a TANGOW log file is described in [Section 4.2 Data Description]. This entry is translated to a format understandable by Weka tools: <*user-id, age, profile, activity, complete, grade, numVisit, action, activityType, success*>. The profile field is actually an aggregated field: entries will contain one value for each attribute in the profile of the related course. In that way, concrete examples of log entries corresponding to the *traffic* course of student *s1* are:

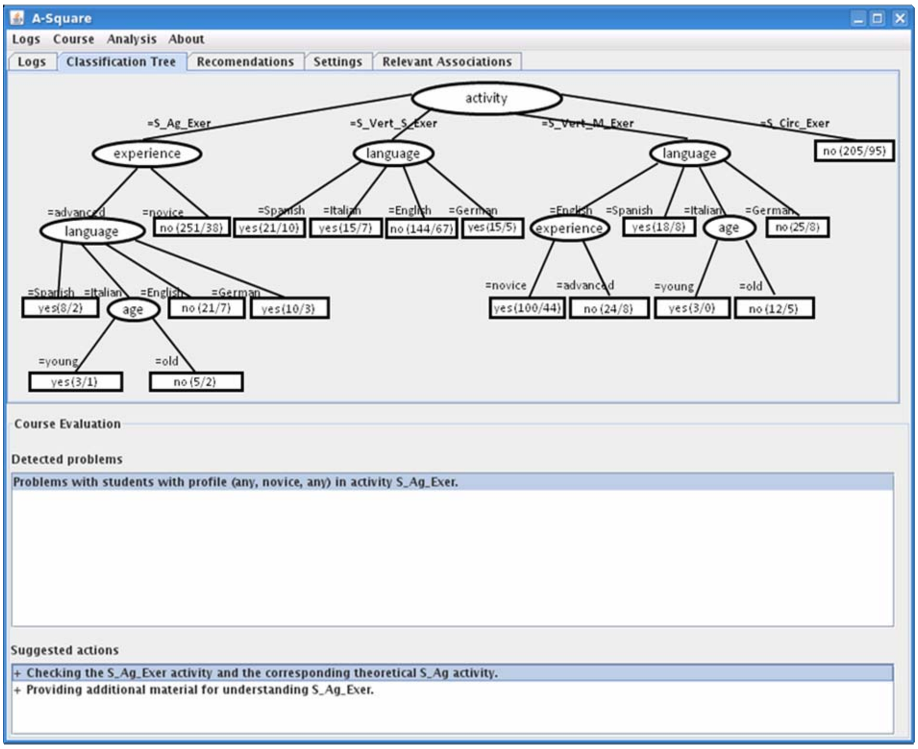
<s1, young, English, novice, S_Ag_Exer, 0.0, 0.0, 1, FIRSTVISIT, P, no>
<s1, young, English, novice,S_Ag_Exer, 0.0, 0.0, 1, LEAVE-ATOMIC, P, no>

These two entries show that student s1 with profile *young*, *English* and *novice* visited the *S_Ag_Exer* activity. It has 0.0 for complete, 0.0 for grade and this is the first visit to this activity. “P” means that this is a practical activity. The second entry shows that the student left the activity *”exercises of signs of traffic policeman”* without completing it (complete = 0.0) and having an insufficient score to pass the exercise; for this reason, success is set to *no*.

Processing: The data is processed with classification algorithms and association rules that produce the results described in the next two subsections.

5.2 Classification Algorithms

When the classification algorithm is applied it generates the tree shown in [Fig. 2].



Detected problems

Problems with students with profile (any, novice, any) in activity S_Ag_Exer.

Suggested actions

+ Checking the S_Ag_Exer activity and the corresponding theoretical S_Ag activity.
+ Providing additional material for understanding S_Ag_Exer.

Figure 2: A^2 user interface, showing the decision tree built up from the logs.

From the interpretation of the decision tree generated by the classification algorithm, it can be concluded that:

- Students that had more problems in practical activity *S_Ag_Exer* were novice. In the tree, they are represented by all the data registers that correspond to novice students that performed practical activity *S_Ag_Exer* at least once. Of these registers, 85%[*] failed. It is worth noting that this percentage corresponds to the number of times that *”exercises of signs of traffic policeman”* was wrongly performed (failed once or more times) by novice students. Therefore it could be considered an alert for the instructor.

- There could be a problem with exercise *S_Circ_Exer (exercises of circular signs)*. However, this situation is not indicated in the tree clearly. In this case, there is 205 times in which students attempted to do the practical activity *S_Circ_Exer*, with 110 failures, representing 54%. This case should not be

[*] The leaf of the decision tree is: no (251/38). This leaf indicates there were 251 instances in which the activity was S_Ag_Exer and experience was novice. The number of instances which had the value “no” in success was 251-38 = 213. Therefore the percentage of failures was 213/251 = 84.86%.

considered a warning because the percentages of success and failure are similar, which makes it impossible to discriminate if there is an abnormal situation.

- The tree shows that students who followed the English version of *traffic* course had some problems with activity *S_Vert_S_Exer (exercises of vertical signs)*. This is evident, since these students attempted to do 144 times the practical activity *S_Vert_S_Exer* failing 77 times. Due to this number of failures represents 54% it should not be considered a warning either.

Tool Recommendations

The tool can give the following recommendations:

- Check the generated content for the practical activity *S_Ag_Exer* for the novice students.
- Check the path taken by a novice student, since the problem (issue) can be located in the previous knowledge of the student.

It is important to consider that the numbers shown on the classification tree (representing the average error of the model) correspond to the results with the synthetic data provided by *Simulog*. When analyzing data generated by real students, the situation will certainly hold more ambiguity. For example, even if the classification tree predicts that student with *novice* profile will fail activity *S_Ag_Exer*, it could be the case that number of students of this profile succeeded in that activity. In data mining terms, the rule would probably not have 100% confidence. The tool will also show this information to the teacher to provide a better understanding of the situation.

It must also be noticed that the number of cases related with the supposed problem are large enough to generate a warning. However, numbers in other cases are not so significant. The only conclusion supported by the evidence is that students with *novice* profile show a clear tendency to have problems on the activity, but nothing can really be said about, for example, the *advance, English* profile. In this direction, more research is currently being carried out with the intention to find empirical thresholds below which no meaningful conclusion can be extracted.

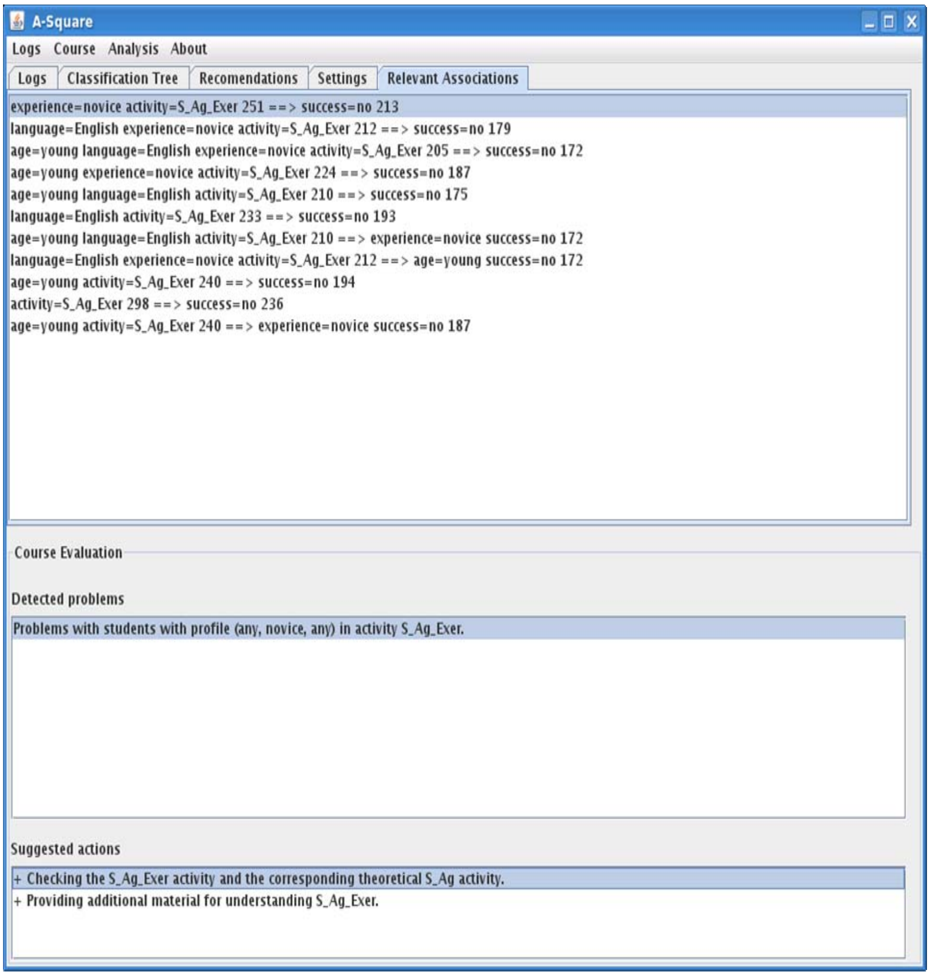
5.3 Association Rules

When association rules are built from the data, the user has to deal with a large list of rules from which the most important ones for the specific application domain must be selected. A^2 implements a filtering mechanism so that only rules which are relevant for evaluating and improving the course are presented to the teacher. The figure 3 shows an example of the type of feedback provided to the teacher.

The selected rule in the [Fig. 3] was the following:

experience=novice activity=S_Ag_Exer 251 → success=no 213

It is read as follows: *novice* students who took any language version of the adaptive course and belong to any age visited the activity *S_Ag_Exer* with failures. Since the right part of the rule is “success = no”. These eleven rules shown in [Fig. 3] were selected automatically from a total of 116 rules.



```
experience=novice activity=S_Ag_Exer 251 ==> success=no 213
language=English experience=novice activity=S_Ag_Exer 212 ==> success=no 179
age=young language=English experience=novice activity=S_Ag_Exer 205 ==> success=no 172
age=young experience=novice activity=S_Ag_Exer 224 ==> success=no 187
age=young language=English activity=S_Ag_Exer 210 ==> success=no 175
language=English activity=S_Ag_Exer 233 ==> success=no 193
age=young language=English activity=S_Ag_Exer 210 ==> experience=novice success=no 172
language=English experience=novice activity=S_Ag_Exer 212 ==> age=young success=no 172
age=young activity=S_Ag_Exer 240 ==> success=no 194
activity=S_Ag_Exer 298 ==> success=no 236
age=young activity=S_Ag_Exer 240 ==> experience=novice success=no 187
```

Problems with students with profile (any, novice, any) in activity S_Ag_Exer.

+ Checking the S_Ag_Exer activity and the corresponding theoretical S_Ag activity.
+ Providing additional material for understanding S_Ag_Exer.

Figure 3: A^2 recommendations extracted from association rules.

Tool Recommendations

The tool can give the following recommendations:

- The instructor must take care about *novice* students of different ages who took any language version of the adaptive course. Since, these students failed the practical activity *S_Ag_Exer* more than twice, because they are bounded to never pass the activity. It is suggested to check the *S_Ag* activity, which is the theoretical activity related to *S_Ag_Exer*. Another solution may be to provide extra material to these students.

6 Tool Validation

In validating the Author Assistant tool one must asses the results gathered from this tool. In other words, the goal of this step is to know if the evaluation tool detected all the symptoms of potential problems inside the interaction data of the students. In this sense, an assisted analysis of the effectiveness of A^2 is possible through Simulog (SIMulation of User LOGs) [Bravo, 06]. Therefore the role of Simulog is to test how the evaluation tool itself works.

6.1 Simulog

Simulog can generate log files imitating the files recorded when a student interacts with the TANGOW system. It 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 course.

Student profiles are generated using a random function that follows a normal distribution, based on user defined parameters. For example, if the Simulog user defines that 70% of the generate logs would correspond to students with language="English" and the remaining 30% with language= "Spanish", and that 200 students will be simulated, Simulog would generate 200 log files (one for each student); the expected value for the total number of students with language="English" is 140.

Simulog mimics the decisions taken by the adaptive system. For example, if the course description contains a rule stating that for "young" students activity A1 is composed by sub activities S1 and S2, which will have to be tackled in this order (S1 before S2), after recording a visit to activity A1 it will record visits to activities S1 and S2, respectively. The user can specify the distribution of every attribute (dimension) defined at the course description, as was described in the above section, and distributions will be combined to generate each student profile. For example, profiles where 90% of the instances have language="English", 90% have experience="novice" and 90% have age="young" can be generated. Therefore, with these values 73% of students with language= "English", experience="novice" and age="young" will be generated on the average.

When two or more activities are available at the same time, the decision about the next visited activity is randomly taken. Values for the time spent by a student in a given activity and how often the student revisits old activities are randomly generated following a Gaussian distribution. These data can also be modified by the user through Simulog parameters.

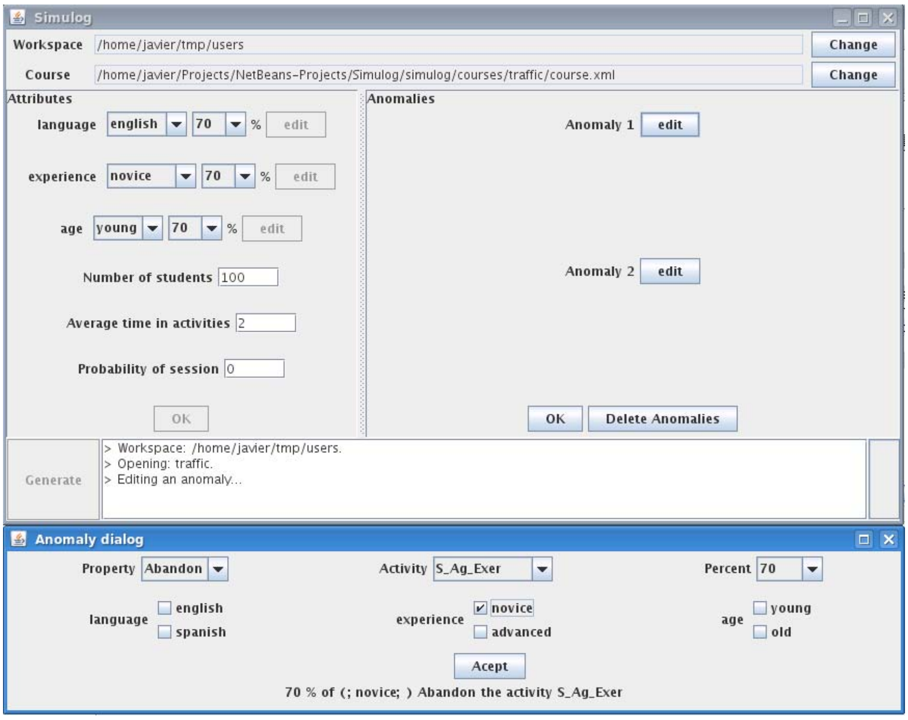
An important feature of Simulog is its ability to generate log files which reflect suspected problems of an adaptive course. For example, if a given course would contain an activity which is particularly difficult for "novice" students, this fact can be reflected on the logs by a significant number of students abandoning the course at that point. An evaluation tool should be capable of finding out this fact through log analysis. In this way, Simulog can be used to generate logs with "controlled" errors, against which an evaluation tool can be tested.

These controlled errors in the logs are called "anomalies" or "symptoms of potential problems", because they potentially reflect unwanted situations on the adaptive course like, for example, most of the student with experience="novice" failing in a given practical task. The Simulog user can specify the anomalies to be represented in the logs. An anomaly is defined by:

- The profile of the simulated students: it describes the scope of the anomaly and is determined by student dimensions. For example, a profile can be: students with language= "Spanish", experience= "advanced" and age="young".
- The corresponding activity: the activity name, for example "S_Ag_Exer".
- The type of anomaly: Are the students failing? Are the students taking to much time? Are the students prematurely abandoning the activity?
- The portion of students with the same profile that will be affected by the anomaly: for example, 60% of the students with the specific profile will fail the test.

In TANGOW an individual log file is generated for each student, and it is composed by three sections: student profile, activities-log and entries-log. Profile contains the dimensions of the student; the activity-log section contains the activities that the student tackled; and the last section contains the student actions while interacting with the course.

Current Simulog implementation is prepared for replicating TANGOW logs. Nevertheless, it is designed with as little dependency on the adaptive system as possible. Therefore, it could be modified to simulate a different AEH system with little effort.



Workspace /home/javier/tmp/users

Course /home/javier/Projects/NetBeans-Projects/Simulog/simulog/courses/traffic/course.xml

language english 70 %

experience novice 70 %

age young 70 %

Number of students 100

Average time in activities 2

Probability of session 0

```
> Workspace: /home/javier/tmp/users.
> Opening: traffic.
> Editing an anomaly...
```

Property Abandon

Activity S_Ag_Exer

Percent 70

language ☐ english ☐ spanish

experience ☑ novice ☐ advanced

age ☐ young ☐ old

70 % of (; novice;) Abandon the activity S_Ag_Exer

Figure 4: Interface of Simulog with Anomaly dialog window activated.

6.2 Using Simulog in validation process

In order to test the Author Assistant capability for detecting problems in the adaptation a validation procedure was carried out.

The first step in the validation procedure was to simulate 100 students by using Simulog. In [Fig. 5] it is shown that firstly the instructor sets the parameters used for the simulation: symptoms of bad adaptation and profiles of students associated with those symptoms. The parameters of the simulation are shown in [Fig. 4]. One symptom of bad adaptation was set *a priori*. Therefore, Simulog generated the interaction data containing the following symptom: 70% of the students with "experience = novice" abandon the activity *S_Ag_Exer* (for more details see section 5.1).

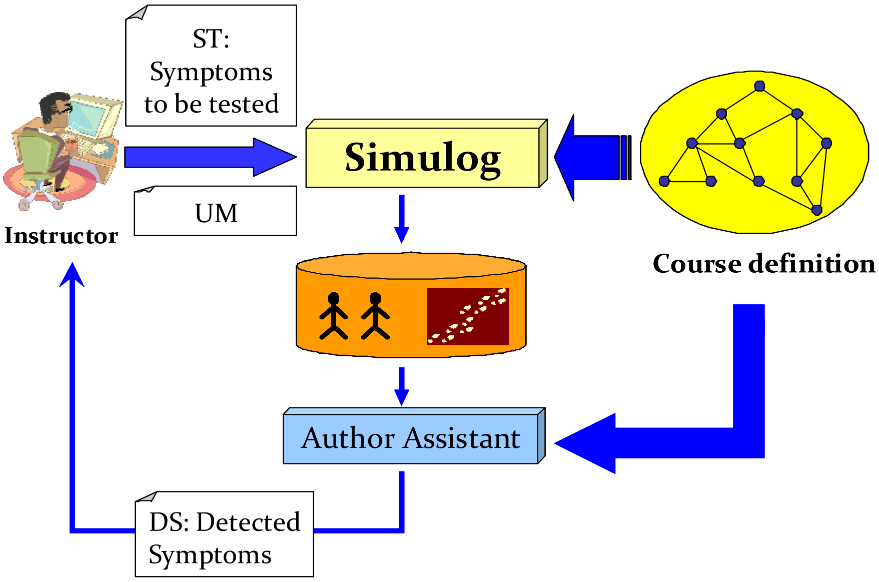


Figure 5: Architecture for validating Author Assistant with Simulog.

The second step was to use Author Assistant with the interaction data generated by Simulog. [Fig. 2 and Fig. 3] show the results of Author Assistant. The [Fig. 2] shows that the decision tree presented 8 leaves with value *no* (failure in the activity) and 9 leaves with value *yes* (success in the activity). However, three of these leaves are only potential symptoms. A^2 only detected one problem, because the percentage of failures (85%) is greater than percentage of successes (15%) in this activity. In addition, Author Assistant does not consider that 54% of failures was significant, because the percentages of failures and successes are similar. For this reason, the other two potential symptoms are not relevant for the evaluation tool (see more details in section 5.2). The same reasoning can be made in [Fig. 3], since A^2 only detected that one association rule was relevant (see more details in section 5.3). Both algorithms got the same symptom: **students that had more problems in the practical activity "exercises of signs of traffic policeman" were novice**.

The final step was to check if Author Assistant was able to detect the symptom of bad adaptation which was set a priori. The initial impression was that A^2 detected the symptom which was generated by Simulog, since profile and activity match on both of them. However, Author Assistant does not provide information about the type of anomaly. Therefore it was unknown if the students abandon this activity or failed it many times.

In conclusion, this use case analysis shows how Author Assistant can obtain relevant results, since it was able to detect the same symptoms of bad adaptation set *a priori*.

7 Analysis, Conclusions and Future work

A proposal for using data mining techniques to support AEH authoring is presented in this paper. In addition, a tool was built to implement this approach. The paper describes the different phases on applying data mining for supporting authors, from the data acquisition and preparation processes to the analysis of the information collected by data mining. More specifically, the use of two techniques is proposed: classification trees and association rules. By using these techniques it is possible to build a model representing the student behavior on a particular course, according to the student profile. This model can be used by teachers to obtain a clear idea of the behavior and performance of student groups within a particular course. While at the same time, this model can also be used as a tool to make overall decisions for a course and/or a whole student group.

The primary goal of the current implementation is to support *a posteriori* analysis of student interactions. Its main benefit is to provide useful information for improving the course to future students. Therefore, two scenarios are possible: courses are periodically offered to different students (for example, once every academic year) or courses permanently available through the Internet.

Our approach is able to provide information of adaptive and non-adaptive E-learning systems, since both systems store information of the student interactions. However, it is especially suitable for providing assistance to AEH authors. When analyzing the logs resulting from 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 learning that more that 80% of the students with profile {"English", "novice"} profile failed the same exercise. In this sense, data mining techniques are able to find patterns in data, separating the wheat from the chaff; in our case, the pattern describes the features of students who failed an exercise.

The approach feasibility is shown by its application within a tool named Author Assistant. This tool is based on data mining algorithms and is able to provide authors advices on how a given adaptive course can be improved. The examples presented are based on the two existing tools: TANGOW and Simulog. Simulog is used in the validation phase for checking whether the supposed errors found by Author Assistant are really reflected on the logs. Moreover, it can also be checked whether the logs reflect additional errors that were not discovered by A^2.

More improvements are needed for Simulog. Though it was developed before the research conducted in this paper, the tool is still in need for adjustment. For example, Simulog capabilities for generating errors (anomalies or symptoms of potential problems) should be improved. In the same way, although Simulog design does not depend on the E-learning system, current implementation does. We are working on providing Simulog independence of specific AEH systems.

Regarding the performance of this data mining approach, we have shown how a problem found by A^2 matches the anomaly established *a priori* in Simulog. This is just a first step on the process of providing advanced support to AEH authoring. However, further work is needed, for example, to be able to extract information about potential problems when only a portion of the students of a given profile show symptoms. The recommendations of the system must also be improved: we expect that more specific recommendations can be provided when the results of two or more techniques are combined.

We are confident that results obtained with this controlled situation can also be mapped to the analysis of logs generated by real users. Currently we are collecting data from real users and further research will be carried out in this direction.

Nevertheless, synthetic logs offer the advantage of fine tuning: logs can be generated to fit exactly any behavior of the evaluation tool that needs to be tested. In that sense, we are currently carrying on experiments to investigate, by mean of heuristic methods, the threshold applicability of each one of the algorithms of data mining used.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Education through project HADA (TIN2007-64716). The first author is also funded by *Fundación Carolina*.

References

[Agrawal, 93] Agrawal, R., Imielinski, T., Swami, A.: “A. Mining association rules between sets of items in large databases”; ACM SIGMOD Proc. of Conference on Management of Data. (1993), 207-216.

[Arabie, 96] Arabie, P., Hubert, J., and De Soete, G.: “Clustering and Classification”; World Scientific Publishing Company / London (1996)

[Bravo, 06] Bravo, J., Ortigosa, A.: “Validating the Evaluation of Adaptive Systems by User Profile Simulation”; Proc. of Workshop on User-Centred Design and Evaluation of Adaptive Systems held at the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), Dublin (2006), 479-483.

[Bravo, 07] Bravo, J., Ortigosa, A., and Vialardi, C.: “A Problem-Oriented Method for Supporting AEH Authors through Data Mining”; Proc. of International Workshop on Applying Data Mining in E-learning (ADML’07) held at the Second European Conference on Technology Enhanced Learning (EC-TEL 2007) (2007), 53-62.

[Becker, 04] Becker, K., Marquardt, C.G., and Ruíz, D.D.: ”A Pre-Processing Tool for Web Usage Mining in the Distance Education Domain”; (2004), 78-87

[Brusilovsky, 03] Brusilovsky, P.: “Developing adaptive educational hypermedia systems: From design models to authoring tools”; In: T. Murray, S. Blessing and S. Ainsworth (eds.): Authoring Tools for Advanced Technology Learning Environment. Dordrecht: Kluwer Academic Publishers; (2003), 377-409.

[Brusilovsky, 98] Brusilovsky, P., Eklund, J., and Schwarz, E.: “Web-based education for all: A tool for developing adaptive courseware” Proc. of 7th Intl World Wide Web Conference, 30 (1-7). (1998), 291-300.

[Carro, 99a] Carro, R.M., Pulido, E., and Rodriguez, P.: “An adaptive driving course based on HTML dynamic generation.”; Proc. of the World Conference on the WWW and Internet WebNet’99, 1, (1999), 171-176.

[Carro, 99b] Carro, R.M., Pulido, E., and Rodríguez, P.: “Dynamic generation of adaptive Internet-based courses”; Journal of Network and Computer Applications, 22, 4. (1999), 249-257.

[Cassidy, 03] Cassidy, S.: “Learning styles: an overview of theories, models and measures”; Proc. Of 8th Annual Conference of the European Learning Styles Information Network (ELSIN), Hull, UK (2003), 419-444

[Chen, 96] Chen, M., Han, J., and Yu, P.: “Data mining: an Overview from Databases Perspective”; IEEE Transactions on Knowledge and Data Engineering, 8, 6 (1996), 866-833.

[Cooley, 99] Cooley, R., Mobasher, B., and Srivasta J.: “Data Preparation For Mining Word Wide Web Browsing Patterns”; Proc. of Knowledge and Information Systems, 1, 1 (1999) 5-32.

[De Bra, 03] De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D., and Stash, N.: “AHA! The Adaptive Hypermedia Architecture”; Proc. of the fourteenth ACM conference on Hypertext and Hypermedia, Nottingham, UK. (2003) 81-84.

[Fayyad, 97] Fayyad, U., Piatetsky-Shapiri, G., and Smyth, P.: “From Data mining to knowledge Discovery in Databases”; AAAI, (1997), 37-54.

[García, 07] García, E., Romero, C., Ventura, S., and Calders, T.: “Drawbacks and solutions of applying association rule mining in learning management systems”; Proc. 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 (2007), 13-22.

[Merceron, 05] Merceron, A., Yacef, K.: “Educational Data Mining: a Case Study”; Proc of the 12th international Conference on Artificial Intelligence in Education AIED, Amsterdam, The Netherlands,IOS Press, (2005), 467-474.

[Merceron, 07] Merceron, A., Yacef, K.: “Revisiting interestingness of strong symmetric association rules in educational data”; Proc. 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), 3-12.

[Mitchell, 97] Mitchell, T.: “Machine Learning”; WCB/McGraw-Hill, Portland (1997)

[Moore, 01] Moore, A., Brailsford, T.J., and Stewart, C.D.: “Personally tailored teaching in WHURLE using conditional transclusion”; Proc of the Twelfth ACM conference on Hypertext and Hypermedia, Denmark (2001) 163-164.

[Pahl, 04] Pahl, C.: “Data Mining Technology for the Evaluation of Learning Content Interaction”; International Journal on E-learning IJEL, AACE, 3, 4 (2004).

[Paredes, 04] Paredes, P., Rodríguez, P.: “A Mixed approach to Modelling Learning Styles in Adaptive Educational Hypermedia”; Proc. of the WBE 2004 Conference. IASTED (2004), 372-378.

[Quinlan, 93] Quinlan, J.R.: “C4.5: Programs for Machine Learning”; Morgan Kaufmann Publishers Inc. / San Francisco, CA (1993).

[Romero, 05] Romero, C., Ventura, S., and Hervás, C.: “Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza basada en web”. Proc. Of III Taller de Minería de Datos y Aprendizaje, Granada, (2005), 49-56.

[Srivastava, 00a] Srivastava J., Cooley R., Deshpande, M., and Tan, P.: “Web Usage Mining: Discovery and Applications of usage Patterns form Web Data”; SIGKDD Explorations, 1,2 (2000), 12-23.

[Srivastava, 00b] Srivastava, J., Mobasher, B., and Cooley, R.:”Automatic Personalization Based on Web Usage Mining”; communications of the Association of Computing Machinery (2000), 142-151.

[Superby, 06] Superby, J.F., Vandamme, J-P., and Meskens, N.: “Determination of factors influencing the achievement of the first-year university students using data mining methods”; Proc. of the Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006), (2006), 37-44.

[Talavera, 04] Talavera, L., Gaudioso, E.: “Mining student data to characterize similar behavior groups in unstructured collaboration spaces” Proc. of workshop on Artificial Intelligence in CSCL. ECAI. (2004), 17-23.

[Witten, 05] itten, I. H., Frank, E.: “Data Mining Practical Machine Learning Tools and Techniques” Morgan Kaufmann Publishers/ San Francisco (2005).

[Zaïane, 01] Zaïane, O.R.: “Web Usage Mining for a Better Web-Based Learning Environment” Proc. of conference on Advanced Technology for Education, Alberta (2001) 60-64.

[Zaïane, 02] Zaïane, O.R.: “Building a Recommender Agent for E-learning Systems”; Proc. of international Conference on Computers in Education, New Zealand (2002), 55-59.

[Zaïane, 06] Zaïane, O.R.: “Recommender system for E-learning: towards non-intrusive web mining of Data mining in E-learning; (Eds. Romero C. and Ventura S.). WitPress, (2006) 79-96.