EDUCATIONAL SOFTWARE EVALUATION: A STUDY FROM AN EDUCATIONAL DATA MINING PERSPECTIVE

Dimitrios P. Lyras [1], Theodor C. Panagiotakopoulos[2], Ilias K. Kotinas[1], Chris T. Panagiotakopoulos[3], Kyriakos N. Sgarbas[1] and Dimitrios K. Lymberopoulos[2]

[1]Artificial Intelligence Group, Wire Communications Laboratory, Electrical & Computer Engineering Department, Greece, GR-26500,
[2]Wire Communications Laboratory, Electrical & Computer Engineering Department, University of Patras, Greece, GR-26500,
[3]Department of Education, University of Patras, Greece, GR-26500

ABSTRACT

In this paper the complicated task of educational software evaluation is revisited and examined from a different point of view. By the means of Educational Data Mining (EDM) techniques, in the present study 177 of the most common evaluation standards that have been proposed by various researchers are examined and evaluated with regards to the degree they affect the effectiveness of educational software. More specifically, via the employment of prediction, feature selection and relationship mining techniques we investigate for the underlying rationale hidden within the data collected from experiments conducted at the Department of Education of the University of Patras with regards to the software evaluation task and the results of this study are presented and discussed in a quantitative and qualitative way.

KEYWORDS

Educational Software Evaluation, Educational Data Mining, EDM, Feature Selection

1. INTRODUCTION

The advent of computers and multimedia technologies during the last years has led to a steadily growing support of the educational task by computers. Nowadays, educational software powered by information and communication technologies seems to illustrate increased capabilities, relatively low cost as well as improved features with regards to the educational task. Along with these developments, there exists a vast increase in the number of educational software provided for use in a class [1], [2].

The term “Educational Software” is used to refer to software designed to support learning [3]. Educational software differentiates to other application software to the fact that during the development of an educational oriented software the way students learn should be taken into account [3], [4]. Moreover, the usability factor of the designed educational software must also be taken under consideration during the development phase since it is tightly connected with the learning process [3], [5] and plays a very important role with regards to the acquisition of the educational software [6], [7].

Nevertheless, as it often occurs with many of the commercial products, educational software is not always suitable and effective in the teaching and learning processes [2]. Awareness regarding the suitability of educational software that has to be used and the kind of assistance that this

DOI : 10.5121/ijma.2014.6301

software has the potential to provide, is very important for a teacher, in order to use educational software in a class [8]. Complementarily, the continuously increasing number of educational software designed and developed for use in class, makes it extremely difficult for the teachers or even for the curriculum experts to decide upon which software, in each specific occasion, is better to use [6]. This is mainly due to the fact that it is very difficult for a teacher to predict the effectiveness of the interaction between the students and the software as well as the learning benefits that the students will gain by using this software, especially in cases where the educational software implements innovative ways to support teaching and learning. This problem is further exacerbated when the teacher is not experienced in the use of information and communication technologies [8].

In cases like the ones described above, the best advisor for the teacher is the evaluation, through which knowledge on the educational value of the examined software is gained. According to Le & Le [2], there are many alternative definitions of the term evaluation. The common basis of all definitions though relies on the fact that evaluation is about assessment of quality of a product, task, program, or activity. The importance of the educational software evaluation process is obvious: on the one hand it allows for deciding upon the most suitable product from a continuously growing variety of software and on the other hand it significantly contributes to promoting software quality and to setting quality standards [9].

According to Scriven [10], there exist two types of software evaluation that are discretized upon the time frame that the evaluation is performed with regards to the software development process:

a) the formative, which is performed during the software development. This type of evaluation is qualitative and focuses on the user.
b) the summative, which is performed after the software development. This type of evaluation is quantitative and focuses on the results of the software implementation and use, with regards to predefined aims set by the development team.

With regards to the later type of software evaluation, it should also be pointed out that it is usually wide scaled and quite demanding in terms of time and space required for the evaluation to be performed. Moreover, since this type focuses at amongst others on the conditions under which the use of the software has a better outcome [11], [12], it is carried out when the software development phase is complete enough allowing thus for the software to be used in real learning situations.

Nevertheless, although most of the criteria used in educational software evaluation are tightly connected with the teaching and learning principles [2], the area of educational software evaluation has been more and more muddled mainly due to the lack of consensus among software evaluators [13], [14], [6]. In particular, there exist many evaluation methods in the research literature spanning from formal or informal to automatic and empirical [15]. The simplest of these methods are very often presented as a list of characteristics that an educator should consider when reviewing a software [16]. Even in such cases though, the validity of such reviews is tied to the expertise and the experience of the reviewer.

With the passing of time the evaluation process becomes more mature, while new techniques, media and data collection methods emerge, continuously enhancing this process [17]. This comes proportionally with the effort that should be put on the evaluation process as it becomes more complicate. Besides the learning effectiveness and the usability, both education software developers and evaluators should monitor many other factors such as software aesthetics, portability, assistance of provided software manuals and guides as well as compatibility with various types of system software and operational systems [6], [4].

In this paper the problem of software evaluation is revisited and examined from an Educational Data Mining (EDM) perspective. According to the Educational Data Mining community website [18], this research area is defined as "an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in".

Taking under consideration the high dimensionality of the software evaluation problem and the lack of consensus among researchers with regards to the criteria used during evaluating an educational software, in the present study we investigate for the correlating/ highly associating factors that significantly affect the effectiveness/quality of an education-oriented software. More specifically, experiments were conducted at the Department of Education of the University of Patras regarding the evaluation of 15 educational software packages approved by the Greek Ministry of Education (GME) for use in class, and the derived data were systematically processed and analyzed via the employment of well-known for their satisfactory performance data mining techniques such as prediction models, feature selection algorithms and relationship mining techniques. The results of this analysis are presented in a quantitative and qualitative way in the following sections. In particular, in Section 2 we provide details regarding the evaluation standards that have been chosen for the needs of the present study, as well as some background information concerning the Educational Data Mining techniques that were employed. In Section 3 a more thorough description of the experimental setup and data collection procedures is provided whereas in Section 4 we thoroughly discuss the experimental results and outline the knowledge that was derived from the systematic processing of the collected data. Finally, we conclude this paper posing our directions for future work.

2. METHODOLOGY

2.1. Evaluation Standards

Regarding the evaluation process several researchers and organizations have proposed hundreds of criteria classified in various categories. The reader may refer to the following researches to receive a more detailed view of these criteria: Preece, & Jones [19], Heller [16], Squires & McDougall [20], Comer & Geissler [21], Georgiadou et al. [22], Belyk & Feist [23], Shaughnessy [24], Panagiotakopoulos, Pierrakeas, & Pintelas, [25], Mihalca [26], Le & Le [2].

However there are some questions that consensus among researchers has not yet been met. For example, which categories should be chosen to perform an educational software evaluation and which criteria of every category should be chosen? According to Wrench [6], with the myriad of different possibilities for areas that should be evaluated, it is easier to see where software evaluators agree on what is effective software than where they disagree. Apart from these problems, the process of software evaluation is expensive and time consuming if done properly [27], [28]. Moreover, how easy can an evaluator answer to an extended list of questions without having bias and errors? When a big number of criteria is used, fatigue may lead the evaluator to wrong assessments and answers [25].

Taking into account all the relevant researches and the criteria selection issues mentioned above, we concluded considering the 177 evaluation criteria which are analytically presented at Appendix 1. These features are classified in the following educational software contexts [25], [29]: Instructional design, User interface, Media and quality of information media, Aesthetics, Content, Navigation, Feedback and interaction, Usability and Ease of Use.

The instructional design context deals with features relevant to: a) the operational framework of a specific educational software, b) its content structure, and c) the determination of the sequence of educational components included in the software.

The features relating to the user interface context describe the intermediate element in the interaction between the user and the system as well as other contextual issues as user-friendliness, design, interoperability etc.

The media and quality of information media context represents the media through which the content of the software is presented (e.g. via the usage of text, images, videos etc.) while at the same time features regarding the assessment of the quality of these media are also considered.

The aesthetics context refers to the employed color schemes and fonts and the uniformity and coherence of the visual distribution.

The content context highlights issues such as included information and projects, information structuring and gnostic fields.

The navigation context is considered in order to evaluate the degree that the software allows the user to easily navigate through the different sections available, facilitating thus the knowledge discovery process.

The feedback and interaction context refers to features relative to the learning effects, the user's level of control over the educational software and the level and the ways that the system improves the learner's cognition with regards to several issues such as the tasks that have to be accomplished, the educational objectives of the available exercises, description of the errors made by the user and so forth.

Finally, the usability context is used to measure the ease of use of the examined educational software and the easiness by which a user may achieve the requested/desired goals.

2.2. Educational Data Mining Techniques and Motivation

As it happens with the closely related Knowledge Discovery in Databases (KDD) field, there exists a wide number of applications of educational data mining. Baker [30] highlights the following four areas of education that have received particular attention within the field:

1. For improving student models that provide an overview of the student's states and characteristics.
2. For discovering or improving models of knowledge structure of the domain (rapid discovery of accurate domain models directly from data).
3. For studying the pedagogical support provided by learning software – discovering which pedagogical support is most effective.
4. For scientific discovery about learning and learners which usually involves application of educational data mining to answer questions in any of the three previously mentioned application areas.

With regards to the different approaches that can be employed in order to achieve the knowledge discovery from educational data, according to Baker [30] they can be categorized as follows: prediction, clustering, relationship mining, discovery with models and distillation of data for human judgment.

For the needs of the present study, the educational data mining approaches that were followed were: prediction, feature selection and association rule mining.

2.2.1. Feature Selection

As described in the previous Section, in the present study 177 of the most common evaluation standards that have been proposed from various researchers were considered. In order to decide upon which software characteristics are most important with regards to the quality of an educational software, feature selection techniques were employed.

Feature selection (aka subset selection or variable selection), is a process where a selection of the available features is performed to be later applied to a learning algorithm [31]. Attribute selection algorithms are often used in order to facilitate data visualization and data understanding, to reduce the measurement and storage requirements as well as the training and utilization times and to defy the curse of dimensionality improving thus the prediction performance [32].

Our main stimulus for concluding on employing feature selection algorithms was three-fold: a) in order to improve the performance of the predictor models, b) to reduce the high dimensionality of the software evaluation problem and therefore provide faster and more cost-effective (in terms of computational time and space requirements) predictors and c) provide a better understanding of the underlying rationale that resulted in the generated data.

2.2.2. Prediction

Data mining techniques that deal with prediction mainly aim at developing models which can infer a single aspect of data (predicted variable) from some combination of other aspects of the data (predictor variables) [30].

In the same publication [30], Baker highlights the two key uses of prediction techniques within the educational data mining field as follows:

1. In order to study what features of a model are important for prediction, giving information about the underlying construct.

2. In order to predict what the output variable would be in contexts where it is not desirable to directly obtain a label for the examined construct.

In the present study, both aspects were taken under consideration. More specifically, benefitting from the powerful probabilistic mechanism of the Bayesian Networks that allows for reasoning under conditions of uncertainty, our experiments aimed at:

i. examining which features (or group of features) results in achieving the highest predictive accuracy and therefore play an important role with regards to the software evaluation task.

ii. constructing a robust and efficient (in terms of accuracy and required computational resources) predictive model that will be able to decide upon the quality/effectiveness of an examined educational software based on several predefined predictor variables. Such a model may appear to be useful as an alternative empirical educational software evaluation system.

With regards to the later goal, it is important to mention that the suggested predictive model is not proposed as a software evaluation system able to come up with the right responses under all circumstances, since it is widely accepted that the software evaluation task is quite complicated and consequently there do not exist such objectively right responses. On the contrary, it should be considered as an alternative assisting tool that might be employed complementarily (or in some cases even substitutionally) with other traditional software evaluation methods.

2.2.3. Relationship Mining

Association rule learning deals with the task of finding interesting associations and/or correlation relationships between variables, in a dataset consisting of a large number of variables. Via the employment of such powerful exploratory techniques, analysts and researchers are able to uncover hidden patterns within the examined datasets.

With regards to the educational data mining field, association rules seem to draw more and more attention by the passing of time [33], [34], [35]. Such techniques are often employed in order to investigate which variables are most strongly associated with a predetermined single variable of particular interest, or in other occasions relationship mining may take the form of attempting to discover which relationships between any two variables are strongest [30]. For the needs of the present study, both directions were followed. In particular, via the employment of Agrawal's Apriori association rule mining algorithm [36], [37]:

i. a series of experiments was conducted in order to investigate for strong relationships between the available variables contained in each one of the evaluation categories presented at Section 2-A (i.e. Educational design, User interface, Media and quality of information media, Aesthetics, Content, Navigation, Feedback and interaction and Usability) and their corresponding overall assessment variables.

ii. an examination for strong associations between any two variables of the collected data was also attempted.

By the means of the aforementioned experimental combinations, our objectives were to:

i. discover strong associations of the form of if-then rules, implying that if some set of variables is encountered, then there exists high probability that another variable will generally also have a specific value.

ii. examine for causal interrelationships between the observed data. In the ideal situation, it would be desirable to know that a rule of the form X→Y does not only imply that these variables correlate with each other, but also suggests that the appearance of X may cause in a way the simultaneous appearance of Y.

With regards to the interestingness criteria used in the present study, the widely accepted measures of support and confidence were employed in combination with the lift criterion, which is considered to be particularly relevant within educational data [38]. Nevertheless, although it is widely accepted that a combination of support, confidence, and either lift or leverage is efficient enough in order to quantitatively measure the "quality" of the rule, the real value of a rule, in terms of usefulness and actionability is subjective and depends heavily of the particular domain and objectives. Therefore, all derived rulesets were also distilled by human experts focusing on the identification of meaningful patterns.

3. EXPERIMENTAL SETUP

The first step of the experimental setup was to determine the educational software packages that would be evaluated. A group of 15 software packages (available at http://www.e-yliko.gr/) were selected, 8 of which are designed for elementary school pupils whereas the remaining 7 were designed for high school pupils. At this point it is important to mention that each software package was designed as an electronic assisting tool covering the educational objectives of the hardcopy material of the respective courses that are being taught in the elementary schools and high schools in Greece.

Consequently, the 8 courses of the elementary school that are being covered by the selected software packages are: Modern Greek language, English language, History, Music, Mathematics, Physics, Religion and Municipal Art. In an equivalent way, the 7 courses of the high school that are being covered by the selected educational software packages are: English language, Chemistry, Philosophy, Biology, Informatics, Homer Epics and Religion.

The next step was to decide upon the group of evaluators that would evaluate the considered educational software packages. The evaluation group consisted of 64 undergraduate students of the Department of Primary Education of the University of Patras, all being potential candidates as teachers for primary (elementary schools) and secondary (high schools) educational institutions.

The software evaluation task took place during the 8th semester of their studies. At that time, they were all attending the course "Computers and Education" and they had already successfully attended the courses "Software Evaluation", "Introduction to Computer Science", and "Networks and Internet". These courses provided them with the necessary skills required in order to evaluate and provide feedback with regards to the considered educational software packages and the features described earlier in this paper.

The data collection and archiving process was performed using an online Educational Software Evaluation Test (ESET hereafter) which was specifically developed in order to cope with the challenges that are imposed during the data collection phase. Our main stimuli for using an online environment instead of employing other conventional offline practices (such as paper and pencil designs) were the facilitation of the data collection process (e.g. easier pooling of individual participant data files) and the incorporation of validation routines as well as threat-avoidance and threat-detection strategies [39] ensuring thus that the information submitted would be in a suitable format and that no questions or selections would have been accidentally missed.

The evaluation procedure had a one-week pilot period through which the online Educational Software Evaluation Test (ESET) was initially filled by members of the research team and a fine-tuning in terms of evaluation criteria semantics was performed. Following this intervention, a group of 6 students was asked to fill the ESET and provide feedback regarding its understandability. According to the received feedback, several features were rephrased in order to further disambiguate their semantics, resulting thus in the final list of evaluated features provided at Appendix 1. The one-week pilot period ended with the release of the final version of the ESET for use by the participants. The total duration of the evaluation period was 2 months ensuring that all the considered software packages would have been evaluated by the 64 participants of our experiments.

4. EXPERIMENTAL RESULTS

The data collected during the experimental phase were systematically processed and analyzed via the employment of data mining techniques. For all the experiments, libraries from the WEKA Machine Learning Workbench [51] have been employed in the form of custom developed Java programs and were properly fine-tuned and parameterized in order to efficiently meet the needs of the present study. The results of this analysis are summarized in a quantitative and qualitative way in the following sections.

4.1. Feature Selection

In order to reduce the high dimensionality of the examined problem and to obtain a more thorough understanding of the way that the generated data were obtained, the Relief-F [40] attribute selection algorithm was employed.

Relief-F, which is an improved and more robust extension of the originally proposed Relief algorithm [41], [42] is a general attribute estimator able to detect conditional dependencies between the examined features. It is widely used as a preprocessing step in classification problems and it has shown good performance in a wide variety of domains [43].

The main reason for concluding on employing the Relief-F algorithm for the needs of the present study was the fact that the educational software evaluation problem involves much feature interaction. In contrast to other heuristic measures for estimating the quality of attributes, Relief algorithms are able to take under consideration the conditional dependence between the attributes and thus are aware of the contextual information and can correctly estimate the quality of attributes in problems with strong dependencies between the attributes [43].

The basic idea of the Relief-F algorithm is to estimate the quality of attributes according to how well their values distinguish between instances that are near to each other. This is achieved by randomly selecting instances, computing their k-nearest neighbors from the same class and the k nearest neighbors from each of the different classes and then adjusting a weighting factor for each feature f according to the formula:

$$w_f = P(diff\ value\ of\ f|diff\ class) - P(diff\ value\ of\ f|same\ class) \tag{1}$$

In the present study, the k parameter, which ensures greater robustness of the algorithm concerning noisy data was set to 10, as proposed by Kononenko in [40]. The evaluation technique used in the present experiments was the 10-cross fold validation technique [44], [45]. In 10-cross fold validation the data are separated in 10 mutually exclusive subsets each one comprising of the same number of instances. Then the evaluation process is performed 10 times, where each time 9 subsets are employed for training of the model and remaining subset is used for its performance measurement. The experimental results averaged over the 10 folds are presented at Table 1. As we may observe from the experimental results presented at Table 1, features 98 and 99 referring to the multimedia characteristics, attributes 42, 49 and 118 referring to the Graphical User Interface (GUI) and aesthetics, and finally features 2, 7, 29, 34, 36, 37 and 138 referring to motivation and learning effectiveness of the examined educational software seem to play the most important role with regards to the educational software evaluation task. These results are in complete accordance with [46] since multimedia are proved to highly affect the degree that information is communicated to the student while at the same time they achieve the arousing of his/ her interest in a variety of ways.

Furthermore, the graphical user interface is considered to be the means enabling the bi-directional communication between the end-user and the system [29]. Consequently, regardless of the interestingness degree of the educational software, in case it is presented within an unattractive and/or dysfunctional GUI, it will probably not fulfil as efficiently the degree of the learning process. Finally, the degree to which the examined software fulfil its educational purposes (i.e. features 2, 36 and 37) are considered particularly critical with regards to the overall assessment of the educational software whereas at the same time educational software that provide motives to the participants achieve more efficiently the engagement of the students to the active learning process and therefore are encouraged as drop-out prevention techniques.

4.2. Prediction

As described at Section 2, in the present study prediction approaches were also taken under consideration in order to determine the set of features that play the most important role with regards to the educational software evaluation task and to develop an efficient predictive model that will be able to decide upon the quality/effectiveness of an examined educational software based on several predefined predictor variables. Towards this direction, the simplest form of Bayesian Networks, i.e. the Naïve Bayes classifier was employed. Given a set of variables (predictors) $X = \{x_1, x_2, \dots, x_n\}$, initially the posterior probability for each class $c_i,\ c_i \in C = \{c_1, c_2, \dots, c_m\}$ is estimated using Bayes' rule:

$$P(c_i|x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n|c_i)P(c_i) \qquad (2)$$

where $P(c_j|x_1, x_2, \dots, x_n)$denotes the probability that X belongs to the categorical level c_j (posterior probability of class membership). But since according to the Naïve Bayes algorithm all variables are mutually independent given a predefined variable, Formula 2 may be further simplified by decomposing the likelihood to a product of terms, as presented at Formula 3.

$$P(c_i|X) = P(c_i)\prod_{k=1}^{n} P(x_k|c_i) \qquad (3)$$

Every new case X can now be classified to the class level c_i that achieves the highest posterior probability. Naïve Bayes is considered to be one of the most efficient and effective inductive learning algorithms within the machine learning and the wider data mining research areas. Despite the conditional independence assumption on which the Naïve Bayes classifier is based, it has been proven to perform surprisingly well in a wide number of applications, including classification [47] and clustering [48] problems. A theoretical explanation of the apparently unreasonable efficacy of Naïve Bayes classifiers is provided at [49].

In the present study, aiming at the identification of the set of features that play the most important role during evaluating educational software, a series of experiments was performed. More specifically, based on the feature ranking results presented at Table 1, 177 Naïve Bayes classifiers were trained, selecting the top-*n* ranked features each time with $n \in \{1,2, \dots 177\}$ (i.e. the first classifier was built only using attribute 99, the second model was constructed using features 99 and 49, the third using features 99, 49 and 98 and so forth). For each classifier, the goodness of the predictor was assessed using Cohen's Kappa statistic evaluation criterion (Cohen, 1960) which is used to measure the agreement between predicted and observed categorizations of a dataset, while correcting for agreement that occurs by chance (Witten & Frank, 2005). Complete agreement corresponds to a Kappa value equal to 1 whereas complete lack of agreement (i.e. purely random coincidences of rates) corresponds to a zero Kappa value. Again, the selected evaluation method was the 10-cross fold validation technique and the corresponding Kappa values for each predictive classifier averaged over the 10 folds are graphically depicted at Figure 1.

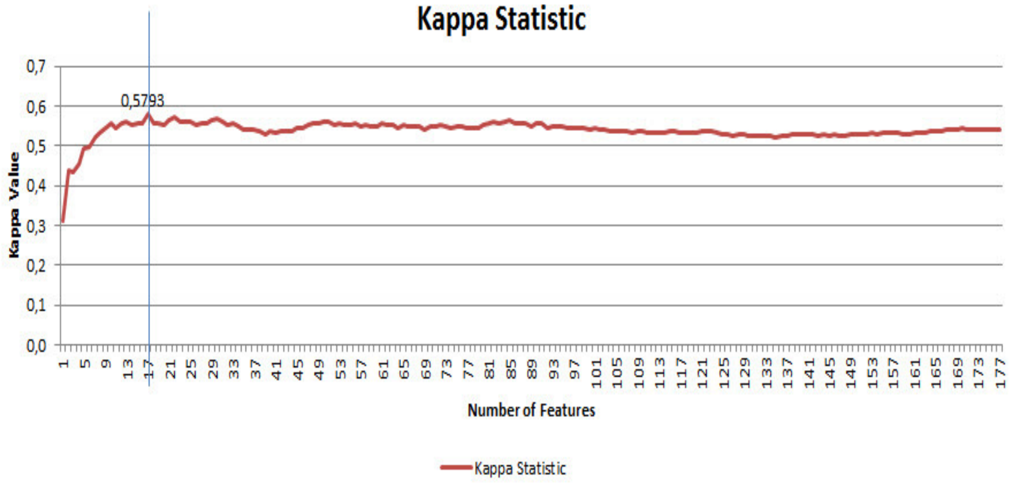


Figure 1. Graphical representation of the Cohen's Kappa Statistic value for the Naïve Bayes classifiers trained over the top-n ranked features of Table I

Table 1. Software Evaluation Features Ranking

	Average Merit	Average Rank	Att.		Average Merit	Average Rank	Att.		Average Merit	Average Rank	Att.
1	0.198 +- 0.006	1.6+- 0.49	99	60	0.079+-0.006	56.9+-9.67	134	119	0.047 +- 0.004	116.4+-8.91	8
2	0.201 +- 0.014	1.6+- 0.8	49	61	0.079 +- 0.006	57.9+-5.37	13	120	0.045 +- 0.003	119.2+-6.13	69
3	0.188 +- 0.006	2.8+- 0.4	98	62	0.078 +- 0.004	58.6+-6.62	11	121	0.044 +- 0.006	120.4+-12.01	131
4	0.168 +- 0.007	4.6+- 0.8	118	63	0.078 +- 0.005	59.4+-6.97	51	122	0.044 +- 0.003	121.6+-7.31	71
5	0.165 +- 0.006	5.9+- 0.94	42	64	0.077 +- 0.006	61.2+-8.84	126	123	0.042 +- 0.003	125.5+-4.57	68
6	0.165 +- 0.007	6.1+- 1.51	7	65	0.076 +- 0.007	61.9+-9.66	1	124	0.041 +- 0.004	126.4+-8.5	77
7	0.164 +- 0.007	6.2+- 1.47	36	66	0.076 +- 0.007	64.9+-3.73	57	125	0.041 +- 0.006	127.3+-14.58	102
8	0.157 +- 0.005	8.9+- 1.64	34	67	0.074 +- 0.004	67.3+-7.13	141	126	0.041 +- 0.005	128.9+-9.79	176
9	0.153 +- 0.007	9.8+- 2.14	2	68	0.072 +- 0.006	68.5+-5.28	52	127	0.04 +- 0.002	129.3+-4.27	72
10	0.151 +- 0.007	10.1+- 1.76	37	69	0.071 +- 0.005	69.2+-9.38	44	128	0.04 +- 0.005	129.6+-10.8	144
11	0.15 +- 0.007	10.5+- 2.01	29	70	0.071 +- 0.007	69.4+-4.88	117	129	0.04 +- 0.005	130.1+-10.95	151
12	0.148 +- 0.006	11.6+- 1.56	138	71	0.07 +- 0.004	70.5+-5.14	47	130	0.039 +- 0.004	130.8+-9	5
13	0.148 +- 0.007	11.8+- 1.66	116	72	0.069 +- 0.004	70.6+-6.8	113	131	0.039 +- 0.005	130.9+-11.09	147
14	0.137 +- 0.007	14.5+- 1.86	101	73	0.07 +- 0.004	71.5+-7.37	162	132	0.039 +- 0.006	131.2+-12.03	103
15	0.131 +- 0.005	16.3+- 1.1	167	74	0.069 +- 0.005	75.8+-6.23	84	133	0.038 +- 0.003	132.6+-6.31	164
16	0.132 +- 0.005	16.3+- 1.95	136	75	0.066 +- 0.004	76.9+-9.45	39	134	0.039 +- 0.005	132.8+-10.67	26
17	0.129 +- 0.005	17.2+- 1.83	108	76	0.066 +- 0.006	77.4+-10.02	160	135	0.038 +- 0.004	132.9+-7.31	24
18	0.127 +- 0.008	17.8+- 3.25	27	77	0.065 +- 0.005	79 +- 8.97	172	136	0.038 +- 0.007	133.9+-12.63	110
19	0.124 +- 0.005	19.6+- 2.06	109	78	0.064 +- 0.006	85.9+- 9.22	79	137	0.038 +- 0.003	135 +-6.75	70
20	0.122 +- 0.004	19.9+-1.51	97	79	0.06 +- 0.004	86.1+- 8.72	65	138	0.037 +- 0.004	136.5+-9.21	105
21	0.12 +- 0.009	21.7+-3.8	35	80	0.06 +- 0.004	86.1+-10.47	128	139	0.037 +- 0.006	136.7+-11.63	174
22	0.117 +- 0.004	22.9+-2.88	95	81	0.061 +- 0.007	86.5+-11.27	168	140	0.037 +- 0.004	136.9+-8.41	22
23	0.115 +- 0.006	25.2+-5.64	140	82	0.06 +- 0.006	86.9+-10.05	85	141	0.036 +- 0.005	138.6+-9.69	41
24	0.114 +- 0.004	25.2+-2.64	90	83	0.059 +- 0.005	87 +- 8.46	129	142	0.036 +- 0.003	139.5+-6.04	67
25	0.112 +- 0.005	25.9+-3.42	92	84	0.06 +- 0.006	87.1+-10.3	173	143	0.035 +- 0.004	139.9+-7.44	64
26	0.112 +- 0.003	26.1+-2.43	145	85	0.06 +- 0.005	87.7+- 7.01	54	144	0.035 +- 0.003	140.8+-7.49	137
27	0.111 +- 0.007	27.4+-5.59	122	86	0.059 +- 0.003	88.9+-10.88	83	145	0.033 +- 0.004	143.7+-6.2	6
28	0.108 +- 0.009	28.9+-5.19	9	87	0.058 +- 0.005	89 +- 8.88	60	146	0.032 +- 0.006	144.7+-8.83	25
29	0.108 +- 0.006	29.1+-3.48	123	88	0.058 +- 0.004	89.2+- 8.12	12	147	0.032 +- 0.004	145.4+-6.41	76
30	0.108 +- 0.005	29.5+-3.98	94	89	0.058 +- 0.004	90.2+-12.11	17	148	0.032 +- 0.005	145.4+-8.71	124
31	0.106 +- 0.005	30.5+-2.62	115	90	0.058 +- 0.006	91.4+-11.68	100	149	0.031 +- 0.005	147.8+-6.65	169
32	0.106 +- 0.005	30.8+-2.75	28	91	0.058 +- 0.005	92.6+-13.92	86	150	0.031 +- 0.003	148.6+-6.68	23
33	0.105 +- 0.009	31.5+-5	177	92	0.058 +- 0.006	93.8+- 7.24	149	151	0.029 +- 0.006	150.8+-9.8	112
34	0.101 +- 0.006	34.4+-3.29	87	93	0.056 +- 0.003	94.4+-11.79	170	152	0.029 +- 0.003	151.3+-6.36	121
35	0.099 +- 0.004	34.7+-3	96	94	0.056 +- 0.005	94.4+-11.75	20	153	0.028 +- 0.006	152.1+-8.69	19
36	0.096 +- 0.007	36.8+-5.55	14	95	0.056 +- 0.004	94.7+- 9.86	171	154	0.028 +- 0.004	153.2+-5.36	33
37	0.095 +- 0.007	39 +-5.81	91	96	0.056 +- 0.004	95.2+- 8.06	62	155	0.027 +- 0.005	155.4+-7.3	111
38	0.094 +- 0.005	39.6+-5.18	89	97	0.056 +- 0.004	95.2+- 9.59	66	156	0.026 +- 0.004	156.5+-5.97	4
39	0.092 +- 0.004	40.3+-3.85	88	98	0.056 +- 0.004	95.3+- 5.29	61	157	0.025 +- 0.004	157.9+-6.38	106
40	0.092 +- 0.006	40.8+-4.49	18	99	0.056 +- 0.002	96 +-10.19	43	158	0.025 +- 0.004	158 +-7.54	45
41	0.091 +- 0.006	43 +-8.19	93	100	0.055 +- 0.005	96.9+-11.78	48	159	0.024 +- 0.006	158.6+-7.8	120
42	0.086 +- 0.005	45.9+-5.58	31	101	0.056 +- 0.006	97.2+-12.16	175	160	0.024 +- 0.003	159.2+-3.63	133
43	0.086 +- 0.005	46.2+-4.81	159	102	0.055 +- 0.005	98.5+- 6.14	75	161	0.024 +- 0.005	159.3+-6.51	38
44	0.087 +- 0.006	46.4+-9.44	82	103	0.054 +- 0.003	98.8+-15.67	73	162	0.023 +- 0.003	160.8+-3.87	139
45	0.086 +- 0.008	46.8+-9.72	16	104	0.055 +- 0.007	100.4+- 8.04	32	163	0.023 +- 0.002	161.3+-3.03	158
46	0.085 +- 0.008	48.5+-8.27	130	105	0.054 +- 0.003	100.6+-10.86	119	164	0.023 +- 0.005	161.5+-6.56	154
47	0.084 +- 0.006	49.6+-6.33	10	106	0.053 +- 0.005	101 +-12.82	53	165	0.021 +- 0.005	163.6+-5.04	74
48	0.083 +- 0.005	50.8+-7.77	152	107	0.053 +- 0.006	107.3+-10.99	78	166	0.019 +- 0.004	165.1+-5.52	163
49	0.084 +- 0.007	51.1+-7.74	3	108	0.051 +- 0.005	108.3+- 6.36	132	167	0.019 +- 0.004	166.3+-3.07	157
50	0.083 +- 0.005	51.4+-7.68	135	109	0.05 +- 0.003	109.3+- 8.32	63	168	0.019 +- 0.002	166.5+-3.5	165
51	0.082 +- 0.004	53 +-6.96	56	110	0.05 +- 0.003	110.4+-10.39	81	169	0.017 +- 0.004	167.7+-4.05	21
52	0.081 +- 0.004	53.5+-6.87	114	111	0.049 +- 0.005	111.3+-15.19	146	170	0.016 +- 0.002	169.3+-2.83	30
53	0.081 +- 0.004	53.6+-6.65	58	112	0.049 +- 0.007	113.8+-10.58	127	171	0.015 +- 0.004	169.9+-4.09	80
54	0.081 +- 0.006	54.6+-7.98	104	113	0.048 +- 0.005	114.4+- 9.49	166	172	0.014 +- 0.006	170.4+-4.18	150
55	0.08 +- 0.005	56.3+-6.26	40	114	0.048 +- 0.005	115.4+- 9.45	55	173	0.013 +- 0.002	171.9+-2.34	161
56	0.079 +- 0.007	56.5+-11.66	142	115	0.047 +- 0.004	115.5+- 7.1	148	174	0.011 +- 0.005	173 +-2.1	153
57	0.079 +- 0.005	56.5+-6.7	59	116	0.047 +- 0.004	115.5+- 8.73	46	175	0.011 +- 0.003	173.7+-1.49	155
58	0.079 +- 0.007	56.7+-8.72	15	117	0.046 +- 0.004	116.1+- 8.7	107	176	0.005 +- 0.005	175.6+-1.69	156
59	0.079 +- 0.004	56.8+-5.25	50	118	0.046 +- 0.004	116.3+-12.82	143	177	0.002 +- 0.002	176.7+-0.46	125

As it can be seen in this figure, there is a significant increase of the predictor's performance after feature 49 is included in the classifier's training set and it continues to increase as more top-ranked features are also considered. This continues to happen until the point where the 17 top-ranked features of Table 1 (i.e. attributes 99, 49, 98, 118, 42, 7, 36, 34, 2, 37, 29, 138, 116, 101, 167, 136 and 108) are employed for the training of the Naïve Bayes classifier and from that point after it can be seen that the inclusion of more features does not result to a further improvement of the predictive model, whereas on the contrary it increases the computational complexity and the resources required in order to induce the classifier. Therefore, according to the experimental results, one could consider the top 17 features of Table 1 as the best trade-off between computational complexity and performance.

Although the usage of the percent accuracy is not usually preferred since it is observed that values of accuracy are highly dependent on the base rates of different classes, in the present study experiments were performed with regards to the predictive accuracy of the induced models in order to examine the direction of employing such predictive classifiers complementarily with other traditional software evaluation methods. Figure 2 graphically depicts the percent accuracy of the Bayesian classifiers when trained over the top-n ranked features of Table 1, with $n \in \{1,2, \dots 177\}$.

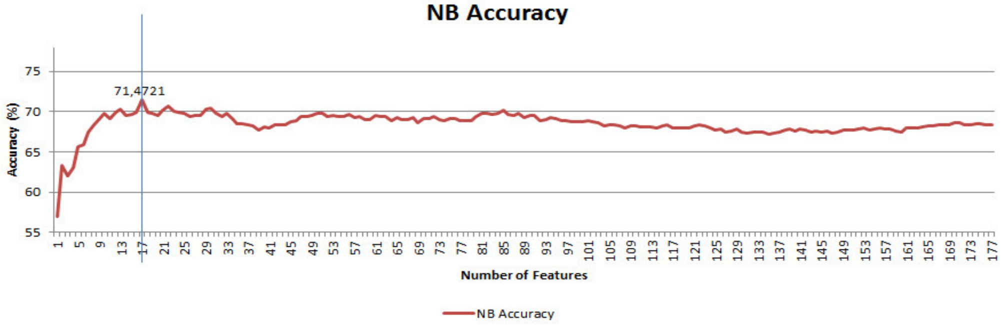


Figure 2. Graphical representation of the percent predictive accuracy for the Naïve Bayes classifiers trained over the top-n ranked features of Table 1

As we may observe, the graph in Figure 2 seems to be in accordance with the one depicted at Figure 1. More specifically, the highest accuracy is observed when the Naïve Bayes classifier is trained over the top-17 features of Table I, achieving the quite satisfactory performance of 71.47% of correctly classified instances, whereas the inclusion of more features in the training set does not contribute to any further improvement of the performance of the predictor. Although, as stated earlier in this paper, the proposed model is not suggested as a software evaluation system able to come up with the right responses under all circumstances, we hope that it might appear to be useful as an automated education software evaluation assisting tool that could be used complementarily (or even substitutionally) with other traditional software evaluation methods in cases where it is not desirable to directly obtain a label for the examined educational software.

The experimental results presented above seem to be in complete accordance with the current state-of-the-art trends with regards to the educational software evaluation task [52]. These results suggest that features examining the usability, functionality and aesthetics of the Graphical User Interface, the degree of the student's participation within the active learning process, the interestingness of the employed multimodal hypermedia modalities and the existence of proper motives supporting the educational objectives seem to be the most prevalent during the educational software evaluation task.

4.3. Relationship Mining

The last direction of the present analysis aimed at the discovery of possible associations and/or causal inter-relationships between the collected data. The relationship mining technique employed for this scope was the Apriori association rule mining algorithm. Apriori is an algorithm proposed by R. Agrawal and R. Shrikant in 1994 [37], which uses a breadth-first search strategy to counting the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. The algorithm uses prior knowledge of frequent itemset properties by employing a level-wise-search where k-itemsets are used in order to explore (k+1)-itemsets [53]. Each rule generated by the Apriori algorithm must satisfy the user-specified thresholds for the support and confidence parameters. The support parameter represents the minimum number of transactions that is required in order to consider a rule as valid; whereas the confidence parameter is used as an indication of the value of the derived rule (i.e. specifies how strong the implication of a rule must be in order to consider the rule valuable).

A problem that may often occur during association rule mining is that of redundant rule generation. Especially when dealing with multidimensional problems such as the one at question it has been observed that the number of generated rules may grow exponentially to the growth of the number of fields contained in the database. In order to reduce the set of rules and causal relationships communicated to the data miner, the most prevalent techniques include the definition of proper minimum support and confidence thresholds with the contemporaneous employment of other interestingness measures such as the lift, the conviction, the leverage and the cosine metrics.

In the present study the problem of redundant rule generation was dealt by setting relatively high thresholds for the support and confidence parameters (0.35 for the support and 0,93 for the confidence parameter) and by employing the lift interestingness measure which has been suggested to be particularly relevant within educational data [38]. Lift, which is expressed as the ration of the confidence of the rule and its expected confidence, represents the degree to which the consequent of an association rule is more likely to be present in the presence of the antecedent of the rule. Lift values greater than 1.0 are highly desirable since they suggest that the antecedent and the consequent appear more often together than expected (i.e. the occurrence of the antecedent has a positive effect on the occurrence of the consequent). Through this analysis, various rules worthy of mentioning were derived. More specifically, with regards to the instructional design of an educationally oriented software it has been observed that in order for a software to fulfil the educational goals for which it was designed (feat. 37), it has to:

i. be comprehensible by users with different learning patterns (feat.1),
ii. assist the learner to concentrate on the object of study (feat. 3)
iii. draw the attention of the learner (feat. 7)
iv. facilitate the learner via its graphical user interface (feat. 31)
v. fulfil the learners' needs (feat. 35) and to
vi. assist the educational process (feat. 36)

Equivalently, the associational analysis of the features regarding the Graphical User Interface of the educational software provided us with rules indicating that the successful GUI design of an educational tool (feat. 49) is highly dependent on:

i. the degree the learner is able to navigate through the software without any further assistance/guidance (feat. 38)
ii. the functionality of the GUI (feat. 40)
iii. the degree the GUI of the examined software follows standard navigation techniques as employed by other well-known and massively used software packages (feat. 45), and on
iv. the degree that the controls and tools of the software are organized and grouped into distinct categories (feat. 47)

The analytic investigation for relational patterns between the features regarding the media and quality of information represented by them, also resulted in the generation of rules worthy of attention. The vast majority of the derived rules suggested that the overall assessment regarding the employment of multimedia in educationally oriented software was higher related to the attractiveness and functionality of the employed graphics and pictures than to other media such as narration, videos, text etc. More specifically, the most prevalent rule reported that the functionality and attractiveness deriving from the usage of multimedia (feat. 99) highly depends on the quality (feat. 87) and attractiveness (feat. 84) of the employed graphics, as well as on the degree these graphics achieve the efficient representation of the educational content (feat. 82) and the degree they facilitate the learner to better comprehend the displayed content (feat. 83).

The experimental results regarding the features referring to the overall aesthetics of the examined educational software reported a strong correlation between the presentation coherence (feat. 117) and the coherence of individual parts forming the displayed image (feat. 104), as well as the harmony of the employed colors (feat. 114 and 115).

In a similar way, the relationship mining experiments regarding the understandability and easiness of comprehension of the displayed educational content (feat. 123) suggested that they are highly affected by:

i. the way the content is displayed, i.e. either uniformly or not (feat 124)
ii. the transparency of the structure and of the categorization of the available information (feat. 131)
iii. the degree the employed media assist the knowledge discovery and acquisition (feat. 138) and
iv. the existence of links to various external educational resources (feat. 144)

Finally, another important rule worthy of mentioning describes a strong relationship between the ease of use of the examined educational software packages (feat. 170) and:

i. the difficulties encountered by the learner on his/her engagement with the software for the first time (feat 168)
ii. the degree a user is in need for further guidance during his/her engagement with the software (feat. 169)
iii. the degree a user was able to anticipate what is need to be done during his/her engagement with the software (feat. 172) and
iv. the ease of use of various tools facilitating the software navigation and information retrieval (feat. 173)

All the aforementioned rules where specifically examined by experts in the field of educational software evaluation and they all concluded to the fact that they provide useful and easy-to-use knowledge with regards to the specific aspects that have to be taken under consideration when designing an education-oriented software.

5. CONCLUDING REMARKS

In the present paper the problem of educational software evaluation is examined from a different point of view. More specifically state-of-the-art Educational Data Mining techniques are employed in order to investigate for the underlying parameters that play the most important role with regards to the software evaluation task.

Towards this direction, experiments were conducted at the Department of Education of the University of Patras regarding the evaluation of 15 educational software approved by GME for use in class. The systematic processing of the derived data revealed useful information about the underlying construct.

In particular, the data analysis using the well-known for its satisfactory performance Relief-F feature selection algorithm suggested that features referring to the multimedia characteristics, the Graphical User Interface (GUI) and the motivation and learning effectiveness of the examined educational software seem to play the most important role with regards to the educational software evaluation task.

Moreover, the collected data were analyzed using prediction techniques. The results of this analysis suggested that the set of the 17 top-ranked features of the feature ranking analysis (i.e. attributes 99, 49, 98, 118, 42, 7, 36, 34, 2, 37, 29, 138, 116, 101, 167, 136 and 108) can be considered as the best trade-off between computational complexity and performance. Furthermore, the Naïve Bayes classifier trained over these features showed a quite satisfactory performance achieving performance equal to 71.47% for the correctly classified instances.

Finally, the relationship mining analysis revealed strong associations between several of the considered features and i) the degree an educational software fulfils its educational goals ii) the success of its GUI design iii) the effectiveness of its overall multimedia usage iv) the overall assessment with regards to the aesthetics of the examined educational software v) the understandability and easiness of comprehension of the displayed educational content and vi) the ease of use of the educational software at question. This mined knowledge might prove to be essential during designing education-oriented software and/or its formative development phase.

REFERENCES

[1] S. Bayram and A. Nous, “Evolution of Educational Software Evaluation: Instructional Software Assessment”, TOJET, vol. 3, no. 2, pp. 21-27, 2004.

[2] Q. Le and T. Le, “Evaluation of Educational Software; Theory into Practice”, Technology and Teaching, pp. 115-124, 2007.

[3] D. Squires and J. Preece, “Usability and Learning: Evaluating the potential of Educational Software”, Computers Educ., vol. 27, no. 1, pp. 15-22, 1996.

[4] C. Ardito et al., “An approach to usability evaluation of e-learning applications”, Univ. Access. Inf. Soc., vol. 4, pp. 270–283, 2006.

[5] A. Jones et al., “Contexts for evaluating educational software”, Interacting with Computers, vol. 11, pp. 499–516, 1999.

[6] J.S. Wrench, “Educational Software Evaluation Form: Towards a New Evaluation of Educational Software”, The Source, vol. 3, no. 1, pp. 34-47, 2001.

[7] S. MacFarlane, G. Sim, and M. Horton, “Assessing Usability and Fun in Educational Software”, in Proceedings of the Conference on Interaction Design and Children, Colorado, 2005, pp. 103-109.

[8] D. Squires and J. Preece, “Predicting quality in educational software: Evaluating for learning, usability and the synergy between them”, Interacting with Computers, vol. 11, pp. 467–483, 1999.

[9] P. Baumgartner and S. Payr, “Methods and Practice of Software Evaluation. The Case of the European Academic Software Award”, in Proceedings of EdMedia 97. Charlotteville: AACE, 1997, pp. 44-50.

[10] M. Scriven, “The methodology of evaluation”, Perspectives of curriculum evaluation, pp. 39–83, 1976.

[11] G. Jacobs, “Evaluating Courseware: Some Critical Questions”, Innovations in Education and Teaching International, vol. 35, no. 1, pp. 3-8, 1998.

[12] A. Karoulis, S. Demetriadis, and A. Pombortsis, “Comparison of expert-based and empirical evaluation methodologies in the case of a CBL environment: the "Orestis" experience”, Computers & Education, vol. 47, pp. 172–185, 2006.

[13] T. Zane and C. G. Frazer, “The extent to which software developers validate their claims”, Journal of Research on Computing in Education, vol. 24, pp. 410-420, 1992.

[14] J. T. Huber and N. B. Guise, “Educational software evaluation process”, Journal of the American Medical Informatics Association, vol. 2, pp. 295-296, 1995.

[15] J. Huart, C. Kolski, and M. Sagar, “Evaluation of multimedia applications using inspection methods: the Cognitive Walkthrough case”, Interacting with Computers, vol. 16, pp. 183–215, 2004.

[16] R. S. Heller, “Evaluating software: A review of the options”, Computers Educ., vol. 17, no. 4, pp. 285-291, 1991.

[17] I. Vlahavas, I. Stamelos, I. Refanidis, and A. Tsoukias, “ESSE: an expert system for software evaluation”, Knowledge-Based Systems, vol. 12, pp. 183–197, 1999.

[18] International Working Group on Educational Data Mining. [Online]. http://www.educationaldatamining.org/index.html

[19] J. Preece and A. Jones, “Training Teachers to Select Educational Computer Software: results of a formative evaluation of an Open University pack”, British Journal of Educational Technology, vol. 16, no. 1, pp. 9-20, 1985.

[20] D. Squires and A. McDougall, “Software evaluation: a situated approach”, Journal of Computer Assisted Learning, vol. 12, pp. 146-161, 1996.

[21] P. Comer and C. Geissler, “A methodology for software evaluation”, in Proceedings of SITE 98: Society for Information Technology & Teacher Education International Conference (9th). Education Resources Information Center, ED421140, 1998.

[22] E. Georgiadou, A. Economides, A. Michailidou, and A. Mosha, “Evaluation of Educational Software Designed for the Purpose of Teaching Programming”, in Proceedings of 9th ICCE SchoolNet 2001 International Conference on Computers in Education, 2001, pp. 745-752.

[23] D. Belyk and D. Feist, “Software evaluation criteria and terminology”, Centre for Distance Education, Online Software Evaluation Report R07/0203 2002.

[24] M. Shaughnessy, “Educational Software Evaluation: A contextual approach”, 2002.

[25] C. Panagiotakopoulos, C. Pierrakeas, and P. Pintelas, “Educational Software and its Evaluation”. Athens: Metaihmio, 2003.

[26] L. Mihalca, “Designing educational software for learning mathematics in primary education”, in Proceedings of Recent Research Developments in Learning Technologies, 2005, pp. 705-716.

[27] H. Geissinger, “Educational Software: Criteria for Evaluation”, in Proceedings of Australian Society for Computers in Learning in Tertiary Education (ASCILITE) 1997, 1997, Retrieved September, 1, 2010 from: http://www.ascilite.org.au/conferences/perth97/papers/Geissinger/Geissinger.html.

[28] T. L. Leacock and J. C. Nesbit, “A Framework for Evaluating the Quality of Multimedia Learning Resources”, Educational Technology & Society, vol. 10, no. 2, pp. 44-59, 2007.

[29] C. Panagiotakopoulos, C. Pierrakeas, and P. Pintelas, “Educational Software Design”. Patras: Hellenic Open University, 2005.

[30] R.S.J. Baker, “Data Mining for Education”, in International Encyclopedia of Education, 3rd ed., B. McGaw, P. Peterson, and E. Baker, Eds.: Oxford, UK:Elsevier, draft available online at http://users.wpi.edu/~rsbaker/publications.html

[31] M. Sewell, “Feature Selection”. [Online]. http://machine-learning.martinsewell.com/feature-selection/

[32] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection”, The Journal of Machine Learning Research, vol. 3, pp. 1157 - 1182 , 2003.

[33] A. Merceron and K. Yacef, “Educational Data Mining: a Case Study”, in Proceedings of Artificial Intelligence in Education (AIED2005), Amsterdam, The Netherlands, 2005.

[34] B. Minaei-Bidgoli, D.A. Kashy, G. Kortemeyer, and W.F Punch, “Predicting student performance: an application of data mining methods with the educational web-based system LON-CAPA”, in ASEE/IEEE Frontiers in Education Conference, 2003.

[35] F. Wang, “On using Data Mining for browsing log analysis in learning environments”, Data Mining in E-Learning. Series: Advances in Management Information, pp. 57-75, 2006.

[36] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”, in Proc. of the ACM-SIGMOD International Conference on the Management of Data, 1993, pp. pp. 207-216.

[37] R. Agrawal and R. Srikant, “Fast Algorithms for mining association rules”, in Proc. of the 20th International Conference On Very Large Databases, 1994, pp. pp. 487-499.

[38] A. Merceron and K. Yacef, “Interestingness Measures for Association Rules in Educational Data”, in Proceedings of the First International Conference on Educational Data Mining, 2008, pp. 57-66.

[39] B. A. Nosek, M. R. Banaji, and A. G. Greenwald, “E-research: Ethics, security, design, and control in psychological research on the Internet”, Journal of Social Issues, vol. 58, no. 1, pp. 161-176, 2002.

[40] I. Kononenko, “Estimating attributes: analysis and extensions of Relief”, in Machine Learning: ECML-94., 1994, pp. 171–182.

[41] K. Kira and L.A. Rendell, “The feature selection problem: traditional methods and new algorithm”, in Proceedings of AAAI’92, 1992a.

[42] K. Kira and L.A. Rendell, “A practical approach to feature selection”, in Machine Learning: Proceedings of International Conference (ICML’92), 1992b, pp. 249–256.

[43] M. Robnik-Sikonja and I. Kononenko, “Theoretical and empirical analysis of ReliefF and RReliefF”, Machine Learning, vol. 53, no. 1-2, pp. 23-69, 2003.

[44] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection”, in Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence , San Mateo, 1995, pp. 1137–1143.

[45] R. Picard and D. Cook, “Cross-Validation of Regression Models”, Journal of the American Statistical Association, pp. 575–583, 1984.

[46] C. Panagiotakopoulos, “Multimedia”, in ‘The Educational Material and the New Technologies’. Patras: Hellenic Open University Publications, 1998.

[47] P. Domingos and M. Pazzani, “On the optimality of the simple Bayesian classifier under zero-one loss”, Machine Learning, vol. 29, pp. 103-130, 1997.

[48] P. Cheeseman and J. Stutz, “Bayesian classification (AutoClass): Theory and results”, In Advances in knowledge discovery and data mining, pp. 153-180, 1996.

[49] H. Zhang, “The Optimality of Naïve Bayes”, in FLAIRS Conference, 2004.

[50] J. Cohen, “A coefficient of agreement for nominal scales”, Educational and Psychological Measurement, vol. 20(1), pp. pp. 37-46, 1960.

[51] I. H. Witten and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques”. San Francisco: Elsevier, 2005.

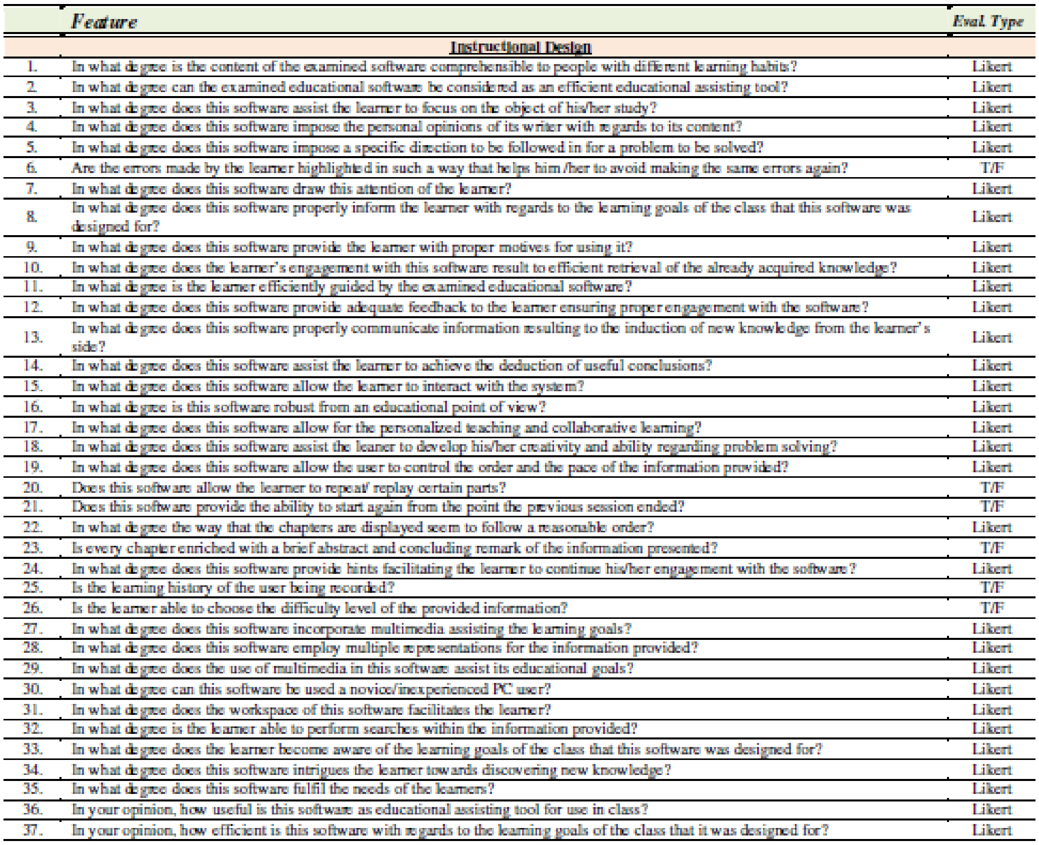
[52] S. Dervan, C. McCosker, B. MacDaniel, and C. O’Nuallain, “Educational multimedia”, Current Developments in Technology-Assisted Education, pp. 801-805, 2006.

[53] J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, 2nd ed.: Morgan Kaufmann, 2006.

APPENDIX 1

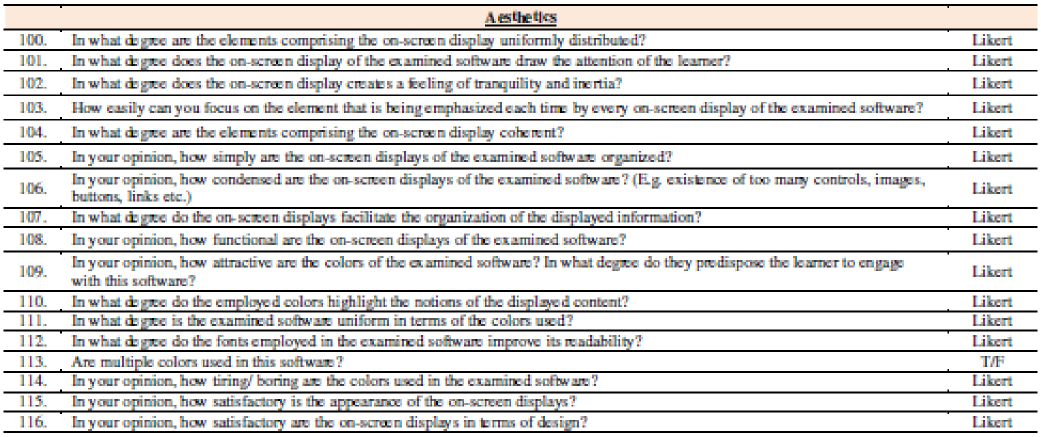
At the following table (Table 2) the 177 features examined in the present study are analytically presented. The evaluation scale employed for 159 of the considered features was the five-level Likert scale, whereas for the remaining ones a True (T) or False (F) answer was taken.

Table 2: Evaluated Features

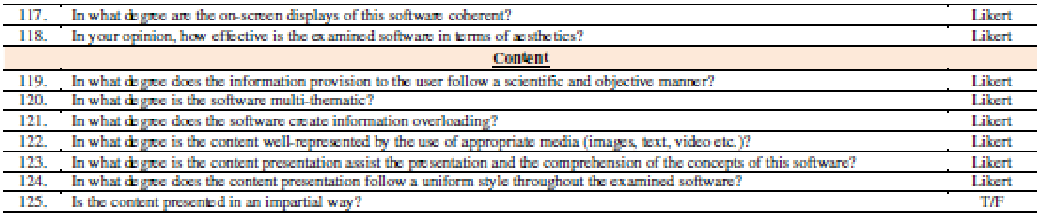


	Feature	*Eval. Type*
	Instructional Design	
1.	In what degree is the content of the examined software comprehensible to people with different learning habits?	Likert
2.	In what degree can the examined educational software be considered as an efficient educational assisting tool?	Likert
3.	In what degree does this software assist the learner to focus on the object of his/her study?	Likert
4.	In what degree does this software impose the personal opinions of its writer with regards to its content?	Likert
5.	In what degree does this software impose a specific direction to be followed in for a problem to be solved?	Likert
6.	Are the errors made by the learner highlighted in such a way that helps him /her to avoid making the same errors again?	T/F
7.	In what degree does this software draw this attention of the learner?	Likert
8.	In what degree does this software properly inform the learner with regards to the learning goals of the class that this software was designed for?	Likert
9.	In what degree does this software provide the learner with proper motives for using it?	Likert
10.	In what degree does the learner’s engagement with this software result to efficient retrieval of the already acquired knowledge?	Likert
11.	In what degree is the learner efficiently guided by the examined educational software?	Likert
12.	In what degree does this software provide adequate feedback to the learner ensuring proper engagement with the software?	Likert
13.	In what degree does this software properly communicate information resulting to the induction of new knowledge from the learner’s side?	Likert
14.	In what degree does this software assist the learner to achieve the deduction of useful conclusions?	Likert
15.	In what degree does this software allow the learner to interact with the system?	Likert
16.	In what degree is this software robust from an educational point of view?	Likert
17.	In what degree does this software allow for the personalized teaching and collaborative learning?	Likert
18.	In what degree does this software assist the leaner to develop his/her creativity and ability regarding problem solving?	Likert
19.	In what degree does this software allow the user to control the order and the pace of the information provided?	Likert
20.	Does this software allow the learner to repeat/ replay certain parts?	T/F
21.	Does this software provide the ability to start again from the point the previous session ended?	T/F
22.	In what degree the way that the chapters are displayed seem to follow a reasonable order?	Likert
23.	Is every chapter enriched with a brief abstract and concluding remark of the information presented?	T/F
24.	In what degree does this software provide hints facilitating the learner to continue his/her engagement with the software?	Likert
25.	Is the learning history of the user being recorded?	T/F
26.	Is the learner able to choose the difficulty level of the provided information?	T/F
27.	In what degree does this software incorporate multimedia assisting the learning goals?	Likert
28.	In what degree does this software employ multiple representations for the information provided?	Likert
29.	In what degree does the use of multimedia in this software assist its educational goals?	Likert
30.	In what degree can this software be used a novice/inexperienced PC user?	Likert
31.	In what degree does the workspace of this software facilitates the learner?	Likert
32.	In what degree is the learner able to perform searches within the information provided?	Likert
33.	In what degree does the learner become aware of the learning goals of the class that this software was designed for?	Likert
34.	In what degree does this software intrigues the learner towards discovering new knowledge?	Likert
35.	In what degree does this software fulfil the needs of the learners?	Likert
36.	In your opinion, how useful is this software as educational assisting tool for use in class?	Likert
37.	In your opinion, how efficient is this software with regards to the learning goals of the class that it was designed for?	Likert

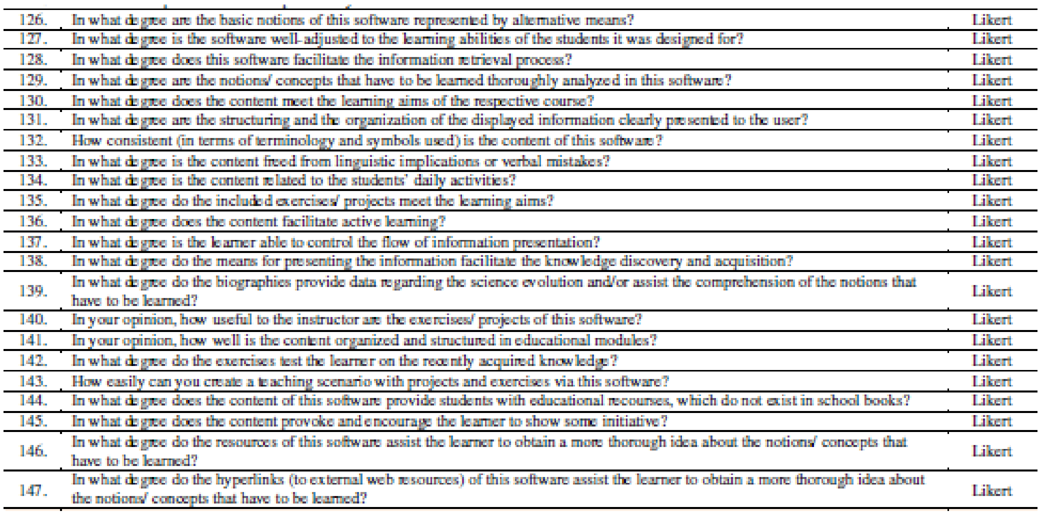
	User Interface	
38.	In what degree can the learner engage himself/ herself with this software without needing any further guidance/support?	Likert
39.	How usable is the user interface of this software?	Likert
40.	In what degree is the user interface of this software functional?	Likert
41.	In your opinion, how simple is the user interface of this software?	Likert
42.	In what degree does the user interface achieve to draw the attention of the user?	Likert
43.	In what degree does this software allow the user to control the way that the information is presented?	Likert
44.	In what degree is the content presentation approved from an educational point of view?	Likert
45.	In what degree the user interface of this software is similar to other well-known and widely used software packages?	Likert
46.	In what degree does this software allow the user to benefit from all the functions of the software?	Likert
47.	In what degree are the tools and controls of this software grouped and organized?	Likert
48.	In your opinion, how functional are the controls of the examined educational software?	Likert
49.	In your opinion, how successful (from an educational point of view) do you consider the user interface of this software?	Likert
	Media and quality of information media	
50.	How satisfactory is the quality of the sound of the examined educational software?	Likert
51.	In what degree does the sound match with the content presented at this software?	Likert
52.	In what degree is the sound uniformly distributed during the learner's engagement with this software?	Likert
53.	In what degree does this software emphasize audibly when the learner starts or ends an activity?	Likert
54.	In what degree does the narrator emphasize on the appropriate spots?	Likert
55.	In what degree are the narration and the way the narrator speaks uniform throughout the examined software?	Likert
56.	In what degree are the graphics and the sound synchronized with each other?	Likert
57.	In your opinion, how well is the sound adjusted with regards to the flow of the presented information?	Likert
58.	In your opinion, how attractive to the learner is the use of sound in this software?	Likert
59.	In what degree does the sound assist the presentation and the comprehension of the content of this software?	Likert
60.	How satisfactory is the quality of the music of the examined educational software?	Likert
61.	In what degree does the music match with the content presented at this software?	Likert
62.	In what degree is the music uniformly distributed during the learner's engagement with this software?	Likert
63.	In your opinion, how well is the music adjusted with regards to the flow of the presented information?	Likert
64.	In what degree does this software achieve emphasis on specific activities via the use of music?	Likert
65.	In your opinion, how attractive to the learner is the use of music in this software?	Likert
66.	In what degree does the music assist the presentation and the comprehension of content of this software?	Likert
67.	How satisfactory is the quality of the videos of the examined educational software?	Likert
68.	In what degree do the videos match with the content presented at this software?	Likert
69.	In what degree do the videos assist the learner to better comprehend the information displayed?	Likert
70.	In your opinion, how tiring is the duration of the videos for the user?	Likert
71.	In your opinion, how attractive to the learner is the use of videos in this software?	Likert
72.	In what degree do the videos assist the presentation and the comprehension of the content of this software?	Likert
73.	In your opinion, how satisfactory are the texts contained in terms of length, content completeness and clarity?	Likert
74.	In what degree is the use of texts excessive in the examined software?	Likert
75.	In your opinion, how readable are the texts?	Likert
76.	In what degree is the employed font style uniform throughout the examined software?	Likert
77.	In what degree is the employed font size uniform throughout the examined software?	Likert
78.	In your opinion, how effective are the texts in this software? (i.e. in what degree do they assist the presentation and the comprehension of the displayed information)	Likert
79.	In your opinion, how satisfactory is the quality of the images in the examined software?	Likert
80.	In what degree is the use of images excessive in the examined software?	Likert
81.	In what degree are the images uniform (in terms of resolution and size) throughout the examined software?	Likert
82.	In what degree do the images match with the content presented at this software?	Likert
83.	In what degree do the images assist the presentation and the comprehension of the content of this software?	Likert
84.	In your opinion, how attractive to the learner is the use of images in this software?	Likert
85.	In what degree do the images assist the presentation and the comprehension of the content of this software?	Likert
86.	In what degree is the use of graphics excessive in the examined software?	Likert
87.	In your opinion, how satisfactory is the quality of the graphics in the examined software?	Likert
88.	In what degree are the graphics uniform throughout the examined software?	Likert
89.	In what degree do the graphics match with the content presented at this software?	Likert
90.	In your opinion, how attractive to the learner is the use of graphics in this software?	Likert
91.	In what degree do the graphics assist the presentation and the comprehension of the content of this software?	Likert
92.	In your opinion, how satisfactory is the quality of the animation in the examined software?	Likert
93.	In what degree is the use of animation excessive in the examined software?	Likert
94.	In what degree is the animation uniform throughout the examined software?	Likert
95.	In what degree does the animation match with the content presented at this software?	Likert
96.	In what degree does the animation assist the presentation and the comprehension of the content of this software?	Likert
97.	In your opinion, how attractive to the learner is the use of animation in this software?	Likert
98.	In what degree does the use of multimedia draw the attention of the learner, assisting at the same time the comprehension of the displayed information?	Likert
99.	In what degree is the use of multimedia attractive to the learner?	Likert



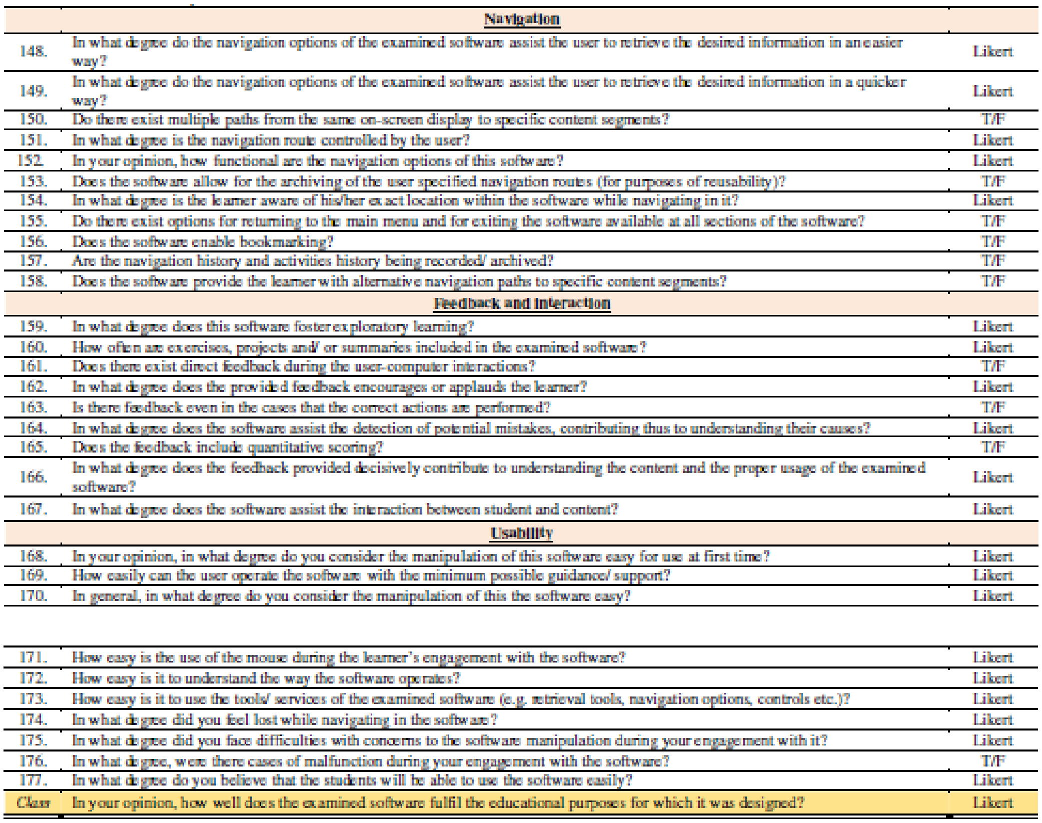
	Aesthetics	
100.	In what degree are the elements comprising the on-screen display uniformly distributed?	Likert
101.	In what degree does the on-screen display of the examined software draw the attention of the learner?	Likert
102.	In what degree does the on-screen display creates a feeling of tranquility and inertia?	Likert
103.	How easily can you focus on the element that is being emphasized each time by every on-screen display of the examined software?	Likert
104.	In what degree are the elements comprising the on-screen display coherent?	Likert
105.	In your opinion, how simply are the on-screen displays of the examined software organized?	Likert
106.	In your opinion, how condensed are the on-screen displays of the examined software? (E.g. existence of too many controls, images, buttons, links etc.)	Likert
107.	In what degree do the on-screen displays facilitate the organization of the displayed information?	Likert
108.	In your opinion, how functional are the on-screen displays of the examined software?	Likert
109.	In your opinion, how attractive are the colors of the examined software? In what degree do they predispose the learner to engage with this software?	Likert
110.	In what degree do the employed colors highlight the notions of the displayed content?	Likert
111.	In what degree is the examined software uniform in terms of the colors used?	Likert
112.	In what degree do the fonts employed in the examined software improve its readability?	Likert
113.	Are multiple colors used in this software?	T/F
114.	In your opinion, how tiring/ boring are the colors used in the examined software?	Likert
115.	In your opinion, how satisfactory is the appearance of the on-screen displays?	Likert
116.	In your opinion, how satisfactory are the on-screen displays in terms of design?	Likert



117.	In what degree are the on-screen displays of this software coherent?	Likert
118.	In your opinion, how effective is the examined software in terms of aesthetics?	Likert
	Content	
119.	In what degree does the information provision to the user follow a scientific and objective manner?	Likert
120.	In what degree is the software multi-thematic?	Likert
121.	In what degree does the software create information overloading?	Likert
122.	In what degree is the content well-represented by the use of appropriate media (images, text, video etc.)?	Likert
123.	In what degree is the content presentation assist the presentation and the comprehension of the concepts of this software?	Likert
124.	In what degree does the content presentation follow a uniform style throughout the examined software?	Likert
125.	Is the content presented in an impartial way?	T/F



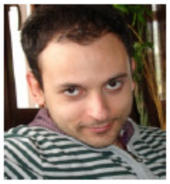
126.	In what degree are the basic notions of this software represented by alternative means?	Likert
127.	In what degree is the software well-adjusted to the learning abilities of the students it was designed for?	Likert
128.	In what degree does this software facilitate the information retrieval process?	Likert
129.	In what degree are the notions/ concepts that have to be learned thoroughly analyzed in this software?	Likert
130.	In what degree does the content meet the learning aims of the respective course?	Likert
131.	In what degree are the structuring and the organization of the displayed information clearly presented to the user?	Likert
132.	How consistent (in terms of terminology and symbols used) is the content of this software?	Likert
133.	In what degree is the content freed from linguistic implications or verbal mistakes?	Likert
134.	In what degree is the content related to the students' daily activities?	Likert
135.	In what degree do the included exercises/ projects meet the learning aims?	Likert
136.	In what degree does the content facilitate active learning?	Likert
137.	In what degree is the learner able to control the flow of information presentation?	Likert
138.	In what degree do the means for presenting the information facilitate the knowledge discovery and acquisition?	Likert
139.	In what degree do the biographies provide data regarding the science evolution and/or assist the comprehension of the notions that have to be learned?	Likert
140.	In your opinion, how useful to the instructor are the exercises/ projects of this software?	Likert
141.	In your opinion, how well is the content organized and structured in educational modules?	Likert
142.	In what degree do the exercises test the learner on the recently acquired knowledge?	Likert
143.	How easily can you create a teaching scenario with projects and exercises via this software?	Likert
144.	In what degree does the content of this software provide students with educational recourses, which do not exist in school books?	Likert
145.	In what degree does the content provoke and encourage the learner to show some initiative?	Likert
146.	In what degree do the resources of this software assist the learner to obtain a more thorough idea about the notions/ concepts that have to be learned?	Likert
147.	In what degree do the hyperlinks (to external web resources) of this software assist the learner to obtain a more thorough idea about the notions/ concepts that have to be learned?	Likert



	Navigation	
148.	In what degree do the navigation options of the examined software assist the user to retrieve the desired information in an easier way?	Likert
149.	In what degree do the navigation options of the examined software assist the user to retrieve the desired information in a quicker way?	Likert
150.	Do there exist multiple paths from the same on-screen display to specific content segments?	T/F
151.	In what degree is the navigation route controlled by the user?	Likert
152.	In your opinion, how functional are the navigation options of this software?	Likert
153.	Does the software allow for the archiving of the user specified navigation routes (for purposes of reusability)?	T/F
154.	In what degree is the learner aware of his/her exact location within the software while navigating in it?	Likert
155.	Do there exist options for returning to the main menu and for exiting the software available at all sections of the software?	T/F
156.	Does the software enable bookmarking?	T/F
157.	Are the navigation history and activities history being recorded/ archived?	T/F
158.	Does the software provide the learner with alternative navigation paths to specific content segments?	T/F
	Feedback and Interaction	
159.	In what degree does this software foster exploratory learning?	Likert
160.	How often are exercises, projects and/ or summaries included in the examined software?	Likert
161.	Does there exist direct feedback during the user-computer interactions?	T/F
162.	In what degree does the provided feedback encourages or applauds the learner?	Likert
163.	Is there feedback even in the cases that the correct actions are performed?	T/F
164.	In what degree does the software assist the detection of potential mistakes, contributing thus to understanding their causes?	Likert
165.	Does the feedback include quantitative scoring?	T/F
166.	In what degree does the feedback provided decisively contribute to understanding the content and the proper usage of the examined software?	Likert
167.	In what degree does the software assist the interaction between student and content?	Likert
	Usability	
168.	In your opinion, in what degree do you consider the manipulation of this software easy for use at first time?	Likert
169.	How easily can the user operate the software with the minimum possible guidance/ support?	Likert
170.	In general, in what degree do you consider the manipulation of this the software easy?	Likert
171.	How easy is the use of the mouse during the learner's engagement with the software?	Likert
172.	How easy is it to understand the way the software operates?	Likert
173.	How easy is it to use the tools/ services of the examined software (e.g. retrieval tools, navigation options, controls etc.)?	Likert
174.	In what degree did you feel lost while navigating in the software?	Likert
175.	In what degree did you face difficulties with concerns to the software manipulation during your engagement with it?	Likert
176.	In what degree, were there cases of malfunction during your engagement with the software?	T/F
177.	In what degree do you believe that the students will be able to use the software easily?	Likert
Class	In your opinion, how well does the examined software fulfil the educational purposes for which it was designed?	Likert

AUTHORS

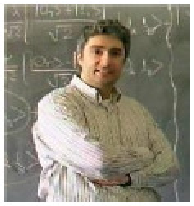
Lyras P. Dimitrios (M '10) was born in Kozani, Greece. In 2006 he graduated from the Electrical and Computer Engineering Department, University of Patras, Greece (field of expertise: Electric Power Systems) and in 2010 he obtained his Ph.D. in the field of Artificial Intelligence from the same department. He speaks fluently English, French, Spanish and Greek, and he is a qualified programmer.



Theodor C Panagiotakopoulos was born in Patras. He took his diploma in Electrical and Computer Engineering from the University of Patras, Greece in 2006. In 2010 he obtained his Ph.D. in the field of bioinformatics and bioengineering in the Department of Electrical and Computer Engineering of the University of Patras. His fields of interest comprise at amongst others: context-awareness, user modeling, emotion recognition, telemedicine systems and applications and biosignals processing.

Ilias Kotinas was born in Athens, Greece. He graduated in February of 2008 from the Electrical Engineering & Computer Technology Department at the University of Patras. Currently, he is a member of the Artificial Intelligence Group of the Laboratory, as a PhD student. His fields of research include Speech Processing, Machine Learning, Data Mining and Text Mining with a focus to the interoperability and integration of the above discrete AI research areas to a common framework.

Chris Panagiotakopoulos was born in Patras, Greece. He received his B.Sc. degree in Mathematics from the University of Ioannina in 1978 and his Ph.D. in the field of Educational Technology from the University of Patras, Greece, in 1997. He is currently an Associate Professor in the Dept. of Primary Education of the University of Patras. His research interests, among others, include design, development, evaluation of educational software, e-learning and open and distance learning. Currently, he is director of Computers and Educational Technology Lab, in the Dept. of Primary Education, University of Patras.



Kyriakos N. Sgarbas (S’89–M’95) was born in Athens, Greece. He received a Diploma (1989) and a Ph.D. (1997) both from the Department of Electrical and Computer Engineering, University of Patras, Greece. He is currently an Assistant Professor at the same Department in Patras University and an Instructor at the School of Science and Technology of the Hellenic Open University. His research interests include Artificial Intelligence, Computational Linguistics, and Quantum Information Processing. Dr. Sgarbas is a member of IEEE, ACM, and several national scientific organizations.

Dimitrios K. Lymberopoulos was born in Tripolis, Greece. He received the Electrical Engineering diploma and the Ph.D. degree from the University of Patras, Greece, in 1980 and 1988, respectively. He is currently a Professor in the Department of Electrical and Computer Engineering, University of Patras. Since 1982, he has been involved as a Technical Supervisor in various research projects funded by the Greek Government, the European Union, the Greek Telecommunication Organization, and the major Greek Telecommunication industries. His research interests include medical communication protocols, telemedicine, context awareness, ontologies in the medical information domain, next generation networks, web multimedia services, data management in medical applications, teleworking (telemedicine) development platforms, and medical communication networks. He is a member of the Technical Chamber of Greece and the Greek Society of Electrical and Mechanical Engineers.