Journal of Universal Computer Science, vol. 10, no. 1 (2004), 47-57
submitted: 29/1/03, accepted: 26/9/03, appeared: 28/1/04 © J.UCS

Discovering Student Models in e-learning Systems[1]

Floriana Esposito
(Dipartimento di Informatica – Università di Bari, Italy
esposito@di.uniba.it)

Oriana Licchelli
(Dipartimento di Informatica – Università di Bari, Italy
licchelli@di.uniba.it)

Giovanni Semeraro
(Dipartimento di Informatica – Università di Bari, Italy
semeraro@di.uniba.it)

Abstract: In all areas of the e-era, personalization plays an important role. Particularly in e-learning a main issue is student modeling, that is the analysis of student behavior and prediction of his/her future behavior and learning performance. In fact, nowadays, the most prevailing issue in the e-learning environment is that it is not easy to monitor students' learning behaviors. In this paper we have focused our attention on the system (the Profile Extractor) based on Machine Learning techniques, which allows for the discovery of preferences, needs and interests of users that have access to an e-learning system. The automatic generation and the discovery of the user profile, to agree as simple student model based on the learning performance and the communication preferences, allow creating a personalized education environment. Moreover, we presented an evaluation of the accuracy of the Profile Extractor system using the classical Information Retrieval metrics.

Key Words: e-learning, learning objects, user context
Categories: K.3.1

1 Introduction

Adaptive personalized e-learning systems could accelerate the learning process by revealing the strengths and weaknesses of each student. They could dynamically plan lessons and personalize communication and the didactic strategy.

Generally, Artificial Intelligence (AI) offers powerful methods, which are useful in the development of adaptive systems. In the past, several intelligent techniques have been experimented in the ITS (Intelligent Tutoring Systems) development: in particular, AI techniques concern the representation of pedagogical knowledge, the construction of the knowledge bases related both to the subject domain and to the didactic strategies and, finally, the student model generation, based on explicit knowledge of the student behavior or on the analysis of the student bugs and misunderstandings. Using AI, Computer-Assisted Instruction systems can be adapted,

---

[1] A short version of this article was presented at the I'KNOW '03 (Graz, Austria, July 2-4, 2003).

during the interaction, to the student personality, characteristics and learning performances.

However, still today, many teaching systems based on the Web have not capitalized such experience and they are often not capable to personalize the instruction material supplied in order to satisfy the needs of each single student. Anyway, a lot of attention has been given to user modeling in e-learning systems: for instance, EUROHELP [Breuker 1990] was devised to provide tools and methods for developing Intelligent Help Systems; InterBook [Brusilovsky and Eklund 1998] provided an user model based on stereotype, which represented the user's knowledge levels about every domain concept, and was modified as the user moved through the information space. Other projects used specific criteria to define a user ability model, e.g. MANIC [Stern et al. 1997], an online courseware system, that determines user typology through heuristics, such as which slides the student has seen and which quizzes he/she has taken.

The main problem is the difference between the concept of user and the concept of student. In a generic web system the user is free to browse and the system attempts to predict the next user steps using the user model to improve the interaction; in the e-learning system the modeling has to improve the educational process, adapting it to the model of the single learner. Therefore it is necessary to control and to assess in some way "student browsing": the student should not be left completely free to make what he/she wants, but must be addressed, through a specific educational path and a continuous evaluation activity of student performance, towards a precise didactic goal. At the moment the evaluation in the e-learning systems, i.e. the constant verification of the training results, is still carried out with traditional multiple-choice questionnaires. The student models, often based on the evaluation of the individual learning benefits during the use of the system and on the student characteristics, are prototypes, due to the difficulty in defining, in terms of explicit knowledge rules, the various behaviors of all the students using the system.

In this paper we propose the development of a component of the e-learning system expressly devoted to the personalization, the Profile Extractor, which allows to automatically discover the user-student preferences, needs and interests and to generate simple student models based on the learning performances and the communication preferences.

Assuming to have a first set of students and to succeed in classifying them in classes, each of which represents a concept (the student category), it is possible, by means of inductive methods of Machine Learning, to infer the concept, i.e. the intentional definitions of student classes, which represent the student models. Data concerning each student is initially collected through preliminary tests to estimate the background knowledge, educational goals, motivation, the preferred modalities of communication etc., and then enriched by the logs of the successive interactions, constitute the training set from which to infer the conceptual user-student models (profiles).

After briefly illustrating the relationships between the user model and the student model, we will introduce some hints concerning the process of automatic extraction of the user/student profiles that can be used in an e-learning system and will evaluate the Profile Extractor accuracy.

2 What is Student Model?

In the area of the Web systems the user models have the task to manipulate information that refer to the knowledge of an user in a specific domain, to his/her personality, his/her preferences, or to any other information that can be useful in the customization of an application.

In the hypermedia educational systems, the student model is the direct extension of the user model and the same techniques to build user models are generally applied in the development of educational material for the assisted instruction [Brusilovsky 1996].

In the area of Intelligent Tutoring Systems, the student model is one of the components to be included in an educational system. In the 1992 Woolf [Woolf 1992] has identified the architecture of an ITS consisting of a set of four major components: the student model, the pedagogical module, the domain knowledge module, and the communication module. In an ITS the student model stores information that is specific to each individual learner: it concerns "how" and "what" the student learns or his/her errors, and the student model plays a main role in planning the training path, supplying information to the pedagogical module of the system. This component provides a pattern of the educational process, using the student model in order to decide the instruction method that reflects the different needs of each student. The domain knowledge module contains information concerning the subject the tutor is teaching, and the communication module creates the interactions with the learner using, through the pedagogical module, the information contained in the student model in order to render the communication more effective. The information collected on the interaction, suitably elaborated, can modify the student model.

On the other side, the use of student models to individualize interaction in hypermedia and on-line instruction systems has been described by several authors [Bull et al. 1995; Bull and Smith 1997; Smith and Jagodzinski 1995], but the application of such techniques to generate effective presentation of instructional material has had little practical success. According to Hartley [Hartley 1998], the root cause is the lack of dialogue between researchers, whereas others believe that it is the complexity of student models [Cummings 1998; Ohlsson 1993; Self 1990].

The range of student modeling approaches available is surveyed by Ragnemalm [Ragnemalm 1996], who distinguishes between models that contain a student's actual domain knowledge and those that contain student characteristics.

In 1996 Vassileva [Vassileva 1996] describes a student model as an example of a general user model, where the student knowledge representation, held in the system, is compared with the domain representation and the expert or desired state representation. The aim of such systems is to compare the student, the domain and the expert models and to attempt to configure information presentation basing upon differences between them, in order to allow the student to reach a desirable knowledge level (educational goal).

In 1996, Brusilovsky [Brusilovsky 1996] faced the problem of developing adaptive hypermedia systems and stated that it is necessary to use some features such as goals, knowledge, background, experience and preferences in order to achieve personalization.

3 Student Modeling in an e-Learning System

In an e-learning application is it necessary to refer to user or student modeling?

The question is not rhetorical: the e-learning is that process of free and irregular learning, but creative and sped up by curiosity, in some involuntary way, generated by the great availability of information on the Web, even whether coming from incoherent sources and in redundant shape. On the other hand, we can define e-learning as a learning process, resulting from the constructive interaction the Web has made possible, the dream of all CAI (Computer Aided Instruction) researchers, which allows to monitor and to improve the educational process, adapting it to the requirements of the single user.

The two meanings are different, the former recalling the spontaneity of the hypermedia browsing lack of control, the latter the requirement of an evaluation process as to the effectiveness and the efficiency of the educational process through a continuous monitoring process. However, it is possible to mediate the two requests trying to model the student as a user in order to improve the interaction, neglecting the problem of monitoring the educational process. The user modeling consists in ascertaining few bits of information about each user, processing that information quickly and providing the results, without the user realizing it. The final result is the construction of the user model or profile that must be differently named: personality profiles, psychographics profiles. The user profiles are, at best, embryonic precursors of an ideal user model, which should possess a deeper and intimate knowledge of the user it refers to. In short, the user model should be able to recognize the user, to know why the user did something, and to foresee what he/she wants to do next. Profiles could be used to deliver personalized content to the user, fitting his/her personal choices.

Such needs are still valid when referring to an e-learning system and to an user who must learn: the possibility to present the instruction material taking into account the preferred or more effective learning strategies or the user personality, the capability of refreshing or recovering concepts, presenting contents in various and attractive shapes in order to improve the attention must be guaranteed.

In the LACAM (Knowledge Acquisition and Machine Learning Laboratory of the University of Bari) a system has been developed to generate user profiles automatically: the Profile Extractor [Abbattista et al. 2002]. This system is a highly reusable module that allows the classification of users through the analysis of past user interaction with the system and employs supervised learning techniques.

Figure 1 shows the complete system architecture, which is further subdivided into four modules: Profile Rules Extractor, Profile Manager, Usage Patterns Extractor and XML I/O Wrapper.

The Profile Manager and the Profile Rules Extractor are the modules mainly involved in the profile generation process; the Usage Patterns Extractor groups dialogue sessions in order to infer some usage patterns that can be exploited for understanding user trends and for grouping single users, who share the same interests and preferences, into user communities [Paliouras et al. 1998]. The XML I/O Wrapper is the layer responsible for the integration of the inner modules with external data sources (using the XML protocol) and for the extraction of the data required for the learning process. The input to the Profile Extractor is represented by the XML file

that contains the personal and interaction data of the user. This information is arranged into a set of unclassified instances, where each instance represents a single user, from the XML I/O Wrapper. The subset of the instances chosen to train the learning system has to be pre-classified by a domain expert (each user is associated with a subset of the categories): this is the actual input to the Profile Rules Extractor, which will infer classification rule sets. The actual user profile generation process is performed by the Profile Manager, on the grounds of the user data and the set of rules induced by the Profile Rules Extractor. When the need to generate/update user profiles arises, the user data are arranged into a set of instances which represents the input to the Profile Manager. On the basis of the classification rule sets inferred, the classifier predicts the user behavior in a system.

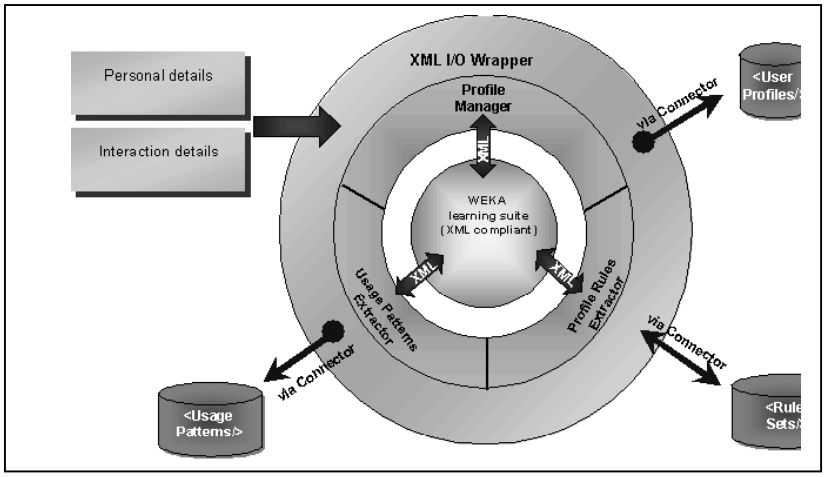


Figure 1: Architecture of the Profile Extractor

For the purpose of extracting user profiles, we focused on supervised machine learning techniques. Starting from pre-classified examples of some target concepts, these techniques induce rules useful for predicting the classification of further unclassified examples. For this reason the core of the Profile Extractor is WEKA [Frank 2000], a machine learning tool developed at the University of Waikato (New Zealand), which provides a uniform interface to many learning algorithms, along with methods for pre/post-processing and for the evaluation of the results of learning schemes, when applied to any given dataset. To integrate WEKA in the Profile Extractor we developed XWEKA, an XML compliant version of WEKA, which is able to represent input and output in XML format. The learning algorithm adopted in the profile generation process is based on PART [Frank and Witten 1998], a rule-based learner that produces rules from pruned partial decision trees, built using C4.5's heuristics [Quinlan 1993]. The antecedent, or precondition, of a rule is a series of tests, just like the tests at nodes in the classification path of a decision tree, while the consequent, or conclusion, gives the class that applies to instances covered by that rule. The main advantage of this method is not performance but simplicity: it produces good rule sets without any need for global optimization.

Extensive experimentation of the system proposed for the automatic extraction of the user profile has been carried out in a field not far from that of e-learning: digital libraries. We experimented with the Profile Extractor System in digital libraries in several contexts like e-Commerce [Abbattista et al. 2002] and contemporary European cultural documents [Licchelli et al. 2003].

The Rules extracted for first experiment: MODULE 1 FUNDAMENTALS COMPUTER SCIENCE

Class: "**GOOD**"

If
NUMER_ACCESS > 17.0
Then Class: Good

Class: "**SUFFICIENT**"

If
INITIAL_SCORE_MODULE_1_SECTION_1 <=0.0 And
FINAL_SCORE_MODULE_1_SECTION_2 <=47.0
Then Class: Sufficient

OR

If
INITIAL_SCORE_MODULE_1_SECTION_1 > 4.0
Then Class: Sufficient

Class: "**INSUFFICIENT**"

If
FINAL_SCORE_MODULE_1_SECTION_2 <= 18.0 And
SCORE_MODULE_1 <= 18.0
Then Class: Insufficient

OR

If
INITIAL_SCORRE_MODULE_1_SECTION_1 > 0.0 And
NUMBER_ACCESS <= 7.0
Then Class: Insufficient

Figure 2: An example of classification rules for the first experiment (Module 1 Fundamentals Computer Science)

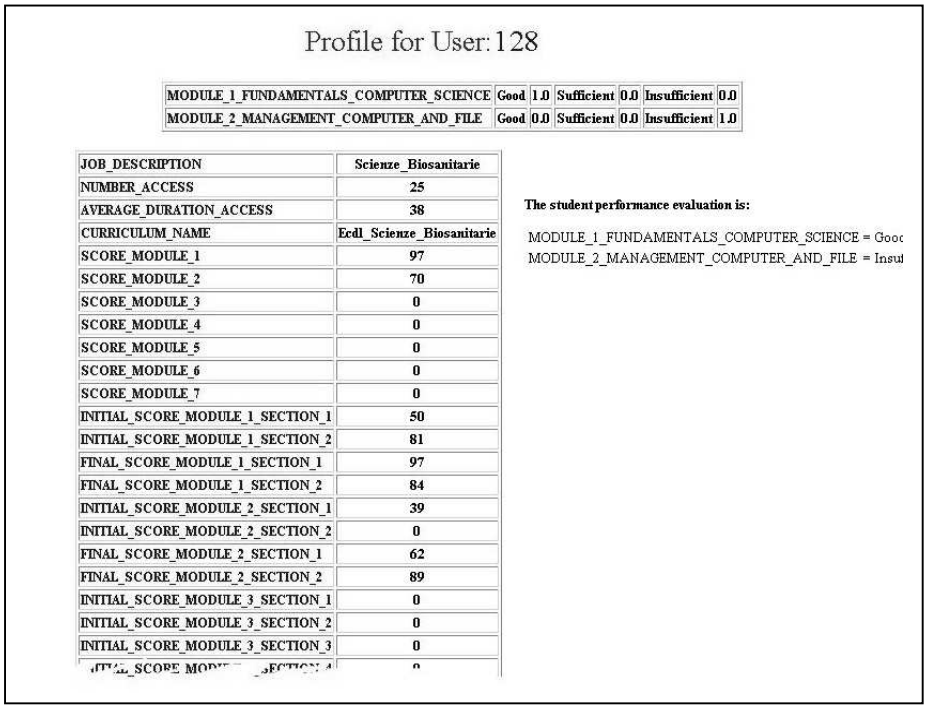
Now, the University of Bari is starting an e-Learning project for a course on Fundamentals of Computer Science for all types of degree (human degree, science degree and etc.). Each student for each kind of degree must attend the first two modules (*Module 1 Fundamentals Computer Science, Module 2 Management Computer and File*), and 3 classes for each experiment (the module) were considered basing upon the final student performance evaluation: good, sufficient or insufficient.

For each class, the system was trained to infer proper classification rules, on the basis of an instance set representing different students. Figure 2 shows the classification rules for the experiment set up on the first module, Module *1 Fundamentals of Computer Science*, on the ground of logs containing interaction and student features; the rule sets may be expressed as disjunctions of conditions.

On the basis of the classification rule sets inferred, the classifier (Profile Manager) can assign a "classification" to new instances (students). In other words, the system predicts whether the user/student is assigned to the classes of performance Good, Sufficient or Insufficient, which are the target classes in experiments. All these classifications, together with the student's details, are gathered to constitute the user-student profiles.

Figure 3 shows an example of a user profile example: the table on the top contains the final classification results as to both the modules, based on the student performance. The detailed data concerning the user appear in the bottom of the table.

These user-student profiles are prototype models useful for managing personalized presentations of the didactic material.


Profile for User: 128

MODULE_1_FUNDAMENTALS_COMPUTER_SCIENCE	Good	1.0	Sufficient	0.0	Insufficient	0.0
MODULE_2_MANAGEMENT_COMPUTER_AND_FILE	Good	0.0	Sufficient	0.0	Insufficient	1.0

JOB_DESCRIPTION	Scienze_Biosanitarie
NUMBER_ACCESS	25
AVERAGE_DURATION_ACCESS	38
CURRICULUM_NAME	Ecdl_Scienze_Biosanitarie
SCORE_MODULE_1	97
SCORE_MODULE_2	70
SCORE_MODULE_3	0
SCORE_MODULE_4	0
SCORE_MODULE_5	0
SCORE_MODULE_6	0
SCORE_MODULE_7	0
INITIAL_SCORE_MODULE_1_SECTION_1	50
INITIAL_SCORE_MODULE_1_SECTION_2	81
FINAL_SCORE_MODULE_1_SECTION_1	97
FINAL_SCORE_MODULE_1_SECTION_2	84
INITIAL_SCORE_MODULE_2_SECTION_1	39
INITIAL_SCORE_MODULE_2_SECTION_2	0
FINAL_SCORE_MODULE_2_SECTION_1	62
FINAL_SCORE_MODULE_2_SECTION_2	89
INITIAL_SCORE_MODULE_3_SECTION_1	0
INITIAL_SCORE_MODULE_3_SECTION_2	0
INITIAL_SCORE_MODULE_3_SECTION_3	0

The student performance evaluation is:

MODULE_1_FUNDAMENTALS_COMPUTER_SCIENCE = Gooc
MODULE_2_MANAGEMENT_COMPUTER_AND_FILE = Insuf

Figure 3: An example of a user profile

4 Measuring the accuracy of Profile Extractor

The main goal of the experiment was to observe the accuracy of the Profile Extractor system in the e-learning field.

For this experiment the data concerning the students enrolled for the online course organized at the University of Bari have been used; the information of each student were gathered in the log file of an e-learning platform.

The experimental dataset contained information on 295 students that were classified, by a domain expert, like *Good*, *Sufficient*, or *Insufficient* student in *MODULE 1 FUNDAMENTALS COMPUTER SCIENCE* and *MODULE 2 MANAGEMENT COMPUTER AND FILE*. The data set was used for the training and the testing phases. As to the composition of the data sets, for module 1 the data are distributed into Good, Sufficient and Insufficient classes with rates of 3% Good - 4% Sufficient - 93% Insufficient while for module 2 the rates are 2% - 1% - 97% respectively. Since the distributions of the data in the classes are so different, of course the experimental results will show the effects of this problem. Indeed the data refer to the first period of the e-learning project and we expect that the student evaluation could be adjusted and refined in operation.

The available data set was used both for the training (90% of the data) and testing phase (10%); the accuracy of the Profile Extractor was measured using a 10-fold cross-validation and several metrics were used in the testing phase. Classification effectiveness has been measured in terms of the classical Information Retrieval (IR) notions of Precision (Pr) and Recall (Re) [Sebastiani 2002].

More in detail, let the classes be {d1 = Good, d2 = Sufficient, d3 = Insufficient}, for each value d_i, the TP (true positive) is the number of test users correctly classified, that is users both the system and the domain expert assigned to class di in the selected experiment. The FP (false positive) is the number of test users incorrectly classified, that is users the system classified as d_i in the selected experiment, differently from the domain expert classification (not d_i) in the same experiment. The FN (false negative) is the number of users incorrectly classified during the test phase, which means the system did not classify users as d_i while the domain expert classified them as d_i.

Then, Recall and Precision are computed as follows:

$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

It is also used F-measure, which is a combination of Precision and Recall:

$$\mathrm{F} = \frac{2 \times \mathrm{Re} \times \mathrm{Pr}}{\mathrm{Pr} + \mathrm{Re}}$$

The experimental results concerning the classification effectiveness are reported for both the experiments: *Module 1 Fundamentals Computer Science (Table 1)* and *Module 2 Management Computer And File (Table 2).*

Class	Pr	Re	F-measure
Good	0.625	0.556	0.588
Sufficient	0.308	0.364	0.333
Insufficient	0.985	0.982	0.984

Table 1: 10-fold cross validation results of the "Module1" experiment

The most important observation from these results is the high accuracy that can be achieved by the system on the Insufficient dataset. The high values of the F1-measure and the balance between recall and precision confirm that the predictions of the Profile Extractor system are accurate, when a high number of training instances for a class is available (class INSUFFICIENT, 93% of 295 students). Of course, if the number of training instance is low, the system produces bad classifications. However the average values of the metrics for this category are sufficiently satisfactory.

Class	Pr	Re	F-measure
Good	0.667	0.800	0.727
Sufficient	0	0	0
Insufficient	1	1	1

Table 2: 10-fold cross validation results of the "Module2" experiment

Also for the second experiment, the results show that a good accuracy can be achieved by the system when the training instances ratio for a class is high (class INSUFFICIENT, 97% of the 295 students). But when the number of the training instance is too much low (class SUFFICIENT, 1% of the 295 students), the system produces incorrect classifications.

Table 3 shows the averages of all experimental results:

Experiment	Avg. Pr	Avg. Re	Avg. F-measure
Module 1	0.639	0.634	0.635
Module 2	0.556	0.6	0.576
Avg.	**0.597**	**0.617**	**0.605**

Table 3: Averages of all 10-fold cross validation results

Values of precision (Pr), recall (Re) and F-measure provide evidence that the system produces sufficiently accurate recommendations if the training set has a good distribution of the examples under the target classes.

5 Future Work

E-learning environments give users a high degree of freedom in following a preferred educational path, together with a control to explore effective paths. This freedom and control is beneficial for the students, resulting in a deeper understanding of the instructional material. Sometimes, this type of e-learning environment is problematic, since some students are not able to explore effectively. One way to address this problem is to augment the environments with personalized support.

Indeed it is possible to adapt an e-learning environment planning a personalized path for each user-student, basing on his needs, goals and characteristics, with the aim of improving the learning process. In this paper, we have focused on student modeling and we have presented a system for automatically generating the profiles of an e-learning user. Once these profiles have been created it is necessary to solve the problem of how to efficiently use such predictive information in order to plan a personalized educational path. Moreover, the student model constructed initially can be refined and/or reviewed on the basis of the new inputs to the system. Once more Machine Learning techniques have turned out to be useful in the automatic refinement of the student models: incremental learning methods are applicable to update the initially acquired knowledge concerning the user, on the basis of new observations.

References

[Abbattista et al. 2002] Abbattista, F., Degemmis, M., Licchelli, O., Lops, P., Semeraro, G., Zambetta, F.: "Improving the usability of an e-commerce web site through personalization"; *Recommendation and Personalization in Ecommerce*. Ricci, F.; Smyth, B. (Eds.). Proc. RPeC'02, Malaga, Spain, (2002), 20-29.

[Breuker 1990] Breuker, J. editor: "EUROHELP: Developing Intelligent Help Systems"; Conceptual model of intelligent help system; EC, Copenhagen. (1990), 41-67.

[Brusilovsky 1996] Brusilovsky, P.: "Methods and techniques in adaptive hypermedia"; User Modelling and User-Adapted Interaction 6, 2-3 (1996), 87-129.

[Brusilovsky and Eklund 1998] Brusilovsky, P., Eklund, J.: "A Study of User Model Based Link Annotation"; Educational Hypermedia. Journal of Universal Computer Science 4, 4 (1998), 429-448.

[Bull et al. 1995] Bull, S., Brna, P,; Pain, H.: "Extending the scope of the student model"; User Modelling and User-Adapted Interaction 5, 10 (1995), 45-65.

[Bull and Smith 1997] Bull, S., Smith, M.: "A pair of student models to encourage collaboration"; Proc. UM97, Italia, (1997), 339-341.

[Cummings 1998] Cummings, G.: "Artificial intelligence in education: an exploration"; Journal of Computer Assisted Learning 14, 4 (1998), 252-259.

[Frank and Witten 1998] Frank, E., Witten, I.H.: "Generating accurate rule sets without global optimization"; Proc. of the 15 th International Conference on Machine Learning. Morgan Kaufmann (1998), 144-151.

[Hartley 1998] Hartley, J.R.: "Ospite Editoriale: CAL and AI - a time for rapprochement?"; Journal of Computer Assisted Learning 14, 4 (1998), 249-250.

[Licchelli et al. 2003] Licchelli, O., Lops, P., Semeraro, G., Bordoni, L., Poggi, F.: "Learning preferences of users accessing digital libraries" ; *Concurrent Engineering – Advanced design, production and management systems.* Cha, J.; Jardim-Gonçalves, R.; Steiger-Garção, A. (Eds.). Proceedings CE'03, Madeira, Portugal, (2003), 457-465.

[Ohlsson 1993] Ohlsson, S.: "Impact of cognitive theory on the practice of authoring"; Journal of Computer Assisted Learning 9, 4 (1993), 194-221.

[Paliouras et al. 1998] Paliouras, G., Papatheodorou, C., Karakaletsis, V., Spyropoulos, C., Malaveta, V.: "Learning User Communities for Improving the Service of Information Providers"; LNCS 1513, Springer (1998), 367-384.

[Quinlan 1993] Quinlan, J.R. : "C4.5: Programs for Machine Learning"; Morgan Kaufmann, San Mateo, CA (1993).

[Ragnemalm 1996] Ragnemalm, E.L.: "Student Diagnosis in Practice; Bridging a Gap"; User Modelling and User-Adapted Interaction 5 (1996), 93-116.

[Sebastiani 2002] Sebastiani, F. (2002) Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34, 1, 1–47.

[Self 1990] Self, J.A.: "Bypassing the intractable problem of student modelling"; Intelligent tutoring systems: at the crossroads of artificial intelligence and education"; Frasson, C., Gauthier, G. (eds.), Ablex Publishing, Norwood, New Jersey (1990), 107-123.

[Smith and Jagodzinski 1995] Smith, C., Jagodzinski, P.: "The implementation of a multimedia learning environment for graduate civil engineers"; Association for Learning Technology Journal 3, 1 (1995), 29-39.

[Stern et al. 1997] Stern, M., Woolf, B.P., Kurose, J.F.: "Intelligence on the Web?" Proc. of the 8th World Conference of the AIED Society, Kobe, Giappone (1997).

[Vassileva 1996] Vassileva, J.: "A task-centred approach for user modelling in a hypermedia office documentation system"; User Modelling and User-Adapted Interaction 6, 2-3 (1996), 185-223.

[Witten and Frank 2000] Witten, I.H., Frank, E.: "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations"; Morgan Kaufmann (2000).

[Woolf 1992] Woolf, B.: "AI in Education"; Encyclopedia of Artificial Intelligence. Shapiro, S. ed., John Wiley & Sons, Inc., New York (1992), 434-444.