

# Multiple language learning: Creating user models through Voronoi iteration

Maria Virvou, Christos Troussas, Efthimios Alepis  
Department of Informatics  
University of Piraeus  
Piraeus, Greece

**Abstract**—Learning foreign languages during the last decade is highly supported by computer systems and their corresponding software applications. However, in the fast and demanding pace of modern life, it is quite common finding people who learn to speak not only one but many foreign languages at the same time. Tutoring systems that support this kind of more demanding electronic teaching need to incorporate intelligence and also adaptation to the needs of their users. Both the creation and the successful incorporation of these user models into a tutoring system are accomplished through the implementation of the Voronoi iteration. In this way, users' characteristics define data points in Euclidean space and are grouped into a given number of categories using k-means clustering. A basic aim of this work is to utilize pre-stored information about past users in order to help future users learn more successfully and more efficiently.

**Keywords**—student modeling; stereotypes; machine learning for user modeling; k-means algorithm; language learning; intelligent tutoring systems

## I. INTRODUCTION

Over the last decade, the use of computer-assisted approaches in the field of education has been growing increasingly. Educational applications are addressed to a very large and heterogeneous audience, namely learners with different characteristics and needs. For this reason, the instruction process should be characterized by high adaptivity and offer students the ability of comprehension and dealing with large cognitive material [14]. Adaptivity is a very crucial issue in educational systems given that there is no teacher available to help users during their learning process [20].

Moreover, our world has witnessed major improvements in the areas of transportation and telecommunications. These important changes have permitted the rise of the phenomenon of globalization by which regional economies, societies, and cultures have become integrated through a global network of people. As a result, all the emerging needs of modern life accentuate the importance of learning foreign languages [16]. Considering the scientific area of Intelligent Tutoring Systems (ITSs), there is an increasing interest in the use of computer-assisted foreign language instruction [11]. In this way, students may learn a foreign language, by using a computer-assisted application. Especially, when these systems offer the possibility of multiple-language learning at the same time, the students may further benefit from this educational process

[12]. The need for tutoring systems that may provide user interface friendliness and also individualized support to errors via a student model are even greater when students are taught more than one foreign languages simultaneously[11]. A solution to this problem may be the integration of the technology of Intelligent Tutoring Systems (ITSs), so as to provide adaptive tutoring to individual students. Educational ITSs are software that aims at providing personalized instruction to students. ITSs offer intelligence and adaptivity to individual students' needs, via student modeling. Adaptive personalized e-learning systems could accelerate the learning process by revealing the strengths and weaknesses of each student. In this sense, these systems could dynamically plan lessons and personalize the communication and didactic strategy [17].

In view of the above, the authors of this paper have implemented an application, based on ITSs, which aims at providing one-on-one tutoring. One of our basic aims has been to promote personalized instruction to students, as our system is designed to know the kind of learner, the learning material and the way of learning. Intelligence and adaptivity are achieved by the incorporation of a student modeling component. The procedure of student modeling is further improved by the incorporation of machine learning techniques for clustering, namely the incorporation of k-means clustering. Building a student model involves defining crucial matters such as the degree of specialization of the students that are modeled, their knowledge and capabilities and also the way of giving assistance, providing feedback and interpreting the behavior of the learner.

This paper is organized as follows. First, we present related work, concerning student modeling in section 2. In section 3, we discuss our system's architecture. Following, in section 3, we present a description of our system, namely a general overview accompanied by screenshots of the system. Finally, in section 4, we come up with a discussion about the usability of the resulting system and we present our future plans.

## II. RELATED WORK

In this section we present related work for student modeling, related firstly with machine learning and secondly with Computer Assisted Language Learning (CALL).

### A. *Student modeling and Machine Learning*

Reference [18] proposed the exploitation of machine learning techniques to improve and adapt the set of user model stereotypes by making use of user log interactions with the system. To do this, a clustering technique is exploited to create a set of user models prototypes; then, an induction module is run on these aggregated classes in order to improve a set of rules aimed as classifying new and unseen users. Their approach exploited the knowledge extracted by the analysis of log interaction data without requiring an explicit feedback from the user. Reference [2] presented a snapshot of what has been investigated in terms of the relationship between machine translation (MT) and foreign language (FL) teaching and learning. Moreover, the author outlined some of the implications of the use of MT and of free online MT for FL learning. Reference [4] investigated which human factors are responsible for the behavior and the stereotypes of digital libraries users so that these human factors can be justified to be considered for personalization. To achieve this aim, the authors have studied if there is a statistical significance between the stereotypes created by robust clustering and each human factor, including cognitive styles, levels of expertise and gender differences. Reference [14] described a web-based educational application for individualized instruction on the domain of programming and algorithms. Their system incorporates a user model, which relies on stereotypes, the determination of which is based on the knowledge level of the learner. Reference [17] focused on machine learning approaches for inducing student profiles, based on Inductive Logic Programming and on methods using numeric algorithms, to be exploited in this environment. Moreover, an experimental session has been carried out from the authors, comparing the effectiveness of these methods along with an evaluation of their efficiency in order to decide how to best exploit them in the induction of student profiles. Reference [20] introduced the ISM framework for the initialization of the student model in Web-based ITSs, which is a methodology that uses an innovative combination of stereotypes and the distance weighted k-nearest neighbor algorithm to set initial values for all aspects of the student model. Reference [5] explained that user modeling poses a number of challenges for machine learning that have hindered its application in user modeling, including the need for large data sets, the need for labeled data, the concept drift and computational complexity. Reference [6] constructed a learning agent that models student behavior at a high level of granularity for mathematics tutor, by using traces from previous users of the tutor to train the machine learning agent.

### B. *Student modeling in CALL*

AutoTutor is a CALL system [1] simulates a human tutor by promoting the conversation and provides feedback to the learner, pumps him/her for more information, gives hints, fills missing information with assertions, identifies and corrects bad answers, answers learner's questions and summarizes answers. Another CALL system is SignMT [15] which tries to

translate sentences/phrases from different sources in four steps, which are word transformation, word constraint, word addition and word ordering. Another computer-based program on second language acquisition is Diglot Reader [3] which is used in a way that students may read a native language text with second language vocabulary and grammatical structures increasingly embedded within the text. TAGARELA is an individualized instruction program [9], which analyzes student input for different activities and provides individual feedback. Finally, Reference [10] described a ubiquitous e-learning tutoring system for multiple language learning, called CAMELL (Computer Assisted Multilingual E-Language Learning). It is a post-desktop model of human-computer interaction in which students "naturally" interact with the system in order to get used to electronically support computer-based learning. Their system presents advances in user modeling, error proneness and user interface design.

However, after a thorough investigation in the related scientific literature, we came up with the result that there was no implementation of multilingual educational systems that combine student modeling and machine learning. Hence, we implemented a prototype system, which incorporates intelligence in its diagnostic component, offers proneness to students' errors provides error diagnosis and advice based on students' needs.

## III. GENERAL ARCHITECTURE OF OUR SYSTEM

The architecture of our system is based on the architecture of ITS architectures. The major functional components of an ITS architecture are the domain model, the student model, the teaching model and the user interface [8].

### A. *Architecture of the Domain Knowledge*

Domain knowledge refers to the curriculum being taught, namely the multiple language learning. The domain knowledge of our multilingual system is consisted of lessons in English and French language. The system follows the same logical structure, for each one of these languages. This structure includes five novice level lessons for beginner students. The first lesson is the learning of the alphabet of the corresponding language. The alphabet is given both in capital letters and in minuscule. The second lesson encompasses the learning of months and days, along with their pronunciation. The third lesson encompasses the genders and the articles, so as to render the students capable of mastering these subjects. The fourth lesson describes in detail the personal and the possessive pronouns. The final lesson familiarizes students with the verbs "to be" and "have", as main verbs. An important issue considering these lessons is that there is a multiple-choice test for each one of the three last lessons, so that the students get evaluated and examined concerning their knowledge and comprehension of the previous lessons. If the students are not found to be adequately prepared to go on to the next lesson, they have to study again the relative theory. We used multiple-choice exercises to evaluate their

performance. The multiple choice questions can be authored by the teacher at any time.

### B. Architecture of the Student Model

Student model provides information about students' state of knowledge along with the way with which they use the system and they learn. The student modeler is responsible for acquiring all the information about the student's performance in the domain as well as his/her proneness to commit errors. The profile of the student, which is constructed by the student model, serves as a source of information that can be used for the interpretation of the student's actions and possible mistakes in solving exercises. The student modeler check the student's answer against the expert's answer and in case of error, it performs error diagnosis. While performing error diagnosis, the student's answer is checked against the set of the erroneous versions that the system is able to identify [12]. One significant source of errors is considered to be the interference between the two languages that the student may learn.

Another crucial task that is performed by the student modeler is the constitution of a history model of the student's weakness and progress. The student model influences the progress of error diagnosis. Students may benefit from viewing their own student models, as they can review their performance and progress each time they want in the educational process. Therefore, this kind of information is used for refining the error diagnosis process and for being presented to the users.

The student modeling mechanism consists of two sub-mechanisms. One mechanism that is responsible for each student's model and another mechanism that reasons about Multilanguage errors (Fig. 1). Each student's profile takes initial values by combining the student's personal information (such as age, gender and educational level) with pre-stored stereotypic information. Consequently, the student model is adapted according to the students' performance while using the educational application. The Multilanguage error diagnosis mechanism tries to find possible reasons about student errors in a Multilanguage domain of knowledge. As a next step, these errors are categorized in terms on five pre-defined categories of errors [13].



Figure 1. Student modeling component

Error diagnosis in the student's domain of knowledge is accomplished by recognizing errors and trying to associate them with one of the following five categories of errors [11]:

i. Article and pronoun mistakes

For example the user may have used "a" instead of "an" or "he" instead of "we".

ii. Spelling mistakes

A spelling mistake is a result of letter redundancy, letter missing or interchange of two neighboring letters.

iii. Verb mistakes

Verb mistakes occur when the user has typed another person than the correct one, for example s/he may have typed "I has" instead of "I have".

iv. Unanswered questions

The user may have no idea about what s/he should write and leave the question unanswered. That means that s/he has lack in theory.

v. Language Confusion

The resulting system is a multilingual learning system, which means that a student may learn two or more languages at the same time. However, there is the possibility of student's getting confused, concerning the proper use of an article or verb.

In order to successfully recognize one or more of the fore mentioned categories of errors, our system incorporates two algorithmic approaches, as illustrated in Fig. 3. The first algorithm tries to find string similarities by matching a student's given "exact" wrong answer with the systems correct stored answer. If string matching occurs in a high percentage the system decides whether the mistake lies between categories i-iv. Correspondingly, using the second algorithm, the system also tries to find meaning similarities between the given and the correct answer by translating these two answers to the system's available supported languages. As an example, the student may have used "I have" instead of "J' ai", which is the French equivalent [13].

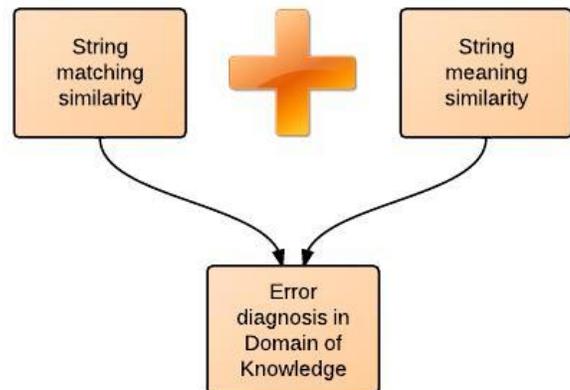


Figure 2. Error Diagnosis component

### C. Teaching Model and User Interface

Teaching model is the advice generator of our system which incorporates the complete way of teaching a curriculum and offers the possibility of personalized help to students during the question answering. The advice generator is activated not only when a student makes an error, but also in every lesson, when the student has to study the theory. Initially, when a student uses the system for the first time, there is only an advice in each lesson, concerning the theory. When the student answers the multiple choice exercises, then s/he is evaluated and gets information from the system in an appropriate way about his/her grade and also is permitted to pass to the next lesson, if his/her performance is satisfactory.

Furthermore, when a student has completed both the theory and the exercises of a lesson in one particular language, s/he has the possibility to do the same lesson in another language. In this way, the system will provide the student with advice, concerning the new theory to be taught and also the differences with the other languages that s/he was taught. User Interface is the interactive environment which stimulates the student's interest in language learning. In addition, it conveys the functionality of a computer-assisted application to the user and translates the user's input into a machine-specific format [7]. The user interface of our system involves all the important components, such as animations and colors, so that it can attract the student's interest in the language learning process and also it can become friendlier to the user.

#### D. K-means for student clustering

The K-mean clustering algorithm is an iterative procedure. In general, the algorithm assigns first an arbitrary initial cluster vector. The second step classifies each vector to the closest cluster. In the third step the new cluster mean vectors are calculated based on all the vectors in this cluster. The second and third steps are repeated until the "change" between the iteration is either quite small or zeroed. The "change" can be defined in several different ways, either by measuring the distances the mean cluster vector have changed from one iteration to another or by the percentage of vectors that have changed between iterations. The objective of the k-means algorithm is to minimize the *within* cluster variability. The objective function (which is to be minimized) is the *sums of squares distances* (errors) between each vector and its assigned cluster center.

$$SS_{distances} = \sum_{\forall x} [x - C(x)]^2 \tag{1}$$

where  $C(x)$  is the mean of the cluster that vector  $x$  is assigned to.

Minimizing the  $SS_{distances}$  is equivalent to minimizing the Mean Squared Error (MSE). The MSE is a measure of the within cluster variability.

$$MSE = \frac{\sum_{\forall x} [x - C(x)]^2}{(N - c)b} = \frac{SS_{distances}}{(N - c)b} \tag{2}$$

where  $N$  is the number of vectors,  $c$  indicates the number of clusters, and  $b$  is the number of spectral bands.

K-means is very sensitive to initial starting values. For two classifications with different initial values and resulting different classification one could choose the classification with the smallest MSE (since this is the objective function to be minimized). However, as we show later, for two different initial values the differences in respects to the MSE are often very small while the classifications are very different. Visually it is often not clear that the classification with the smaller MSE is truly the better classification. From a statistical viewpoint, the clusters obtained by k-mean can be interpreted as the Maximum Likelihood Estimates (MLE) for the cluster

means if we assume that each cluster comes from a spherical Normal distribution with different means but identical variance (and zero covariance).

#### E. Parameterization of K-means algorithm

The  $k$ -means clustering algorithm is classically described as taking an input set  $x$  of  $m$  items,  $x_1, x_2, \dots, x_m$ , where each item  $x_i$  has some corresponding vector  $\psi_i \in \mathbb{R}^N$ . A clustering algorithm computes some clustering  $y$  of  $x$  with  $k$  clusters so as to minimize intracluster Euclidean distance over these  $\psi_i$ , i.e.,

$$\operatorname{argmin}_y \sum_{c \in \mathbf{y}} \sum_{x_i \in c} \left\| \psi_i - \frac{\sum_{x_j \in c} \psi_j}{|c|} \right\|_2^2 \tag{3}$$

Algebraic manipulation reveals this minimization is equivalent to finding  $y$  to maximize

$$\operatorname{argmax}_y \sum_{c \in \mathbf{y}} \frac{1}{|c|} \sum_{i, j \in c} \langle \psi_i, \psi_j \rangle \tag{4}$$

in a form often called kernel  $k$ -means.

How can we parameterize this (4) objective function to provide a family of similarity measures for learning? A simple but powerful parameterization is to provide some linear weighting  $w \in \mathbb{R}^N$  to distort the  $\psi_i$  dimensions:

$$\operatorname{argmax}_y \sum_{c \in \mathbf{y}} \frac{1}{|c|} \sum_{i, j \in c} \psi_i^T \operatorname{diag}(w) \psi_j \tag{5}$$

We can alternately phrase (5) as

$$\operatorname{argmax}_y \sum_{c \in \mathbf{y}} \frac{1}{|c|} \sum_{i, j \in c} \langle w, \psi_i \circ \psi_j \rangle \tag{6}$$

In the equation above, "o" is the componentwise vector product. By changing weights in  $w$ , we affect what clustering  $y$  of  $x$  is optimal under this parameterized  $k$ -means objective (6) [19].

The aim of clustering analysis is to group data in such a way that similar objects are in one cluster and objects of different clusters are dissimilar. The k-Means algorithm basically consists of the three aforementioned steps:

1. an initial set of 'k' (where 'k' is the number of clusters) so-called centroids, i.e. virtual points in the data space is randomly created,
2. every point of the data set is assigned to its nearest centroid and
3. the position of the centroid is updated by the means of the data points assigned to that cluster. In other words, the centroid is moved toward the center of its assigned points.

This is done until no centroid was shifted in one iteration resulting in 'k' subsets/cluster.

The assignment of points to centroids results in a partitioning of the data space. It results in virtual borders between each two centroids, where the distance is equal on each side of the border to these centroids. This kind of partitioning is also known as a Voronoi tessellation.

In this small animation a grid of points is created with each vector as one point and 'k' randomly created centroids then the k-Means runs until convergence. After convergence the centroids are again randomly initialized and the k-Means runs again and so on. Since every possible point of the input space is assigned to one centroid the result is a Voronoi tessellation of the input space.

For the incorporation of the algorithm into the resulting multilingual system (Fig. 3) we may observe the following basic steps:

- i. For the initialization of the system the k-means algorithm receives as input, pre-stored data or data from empirical studies. It uses two fundamental characteristics which tend to influence the educational procedure. Namely, the age of students along with their level of knowledge in one of the foreign language taught. These characteristics have been found quite significant in past language learning applications [20].
- ii. Machine learning techniques are used as a next step in order to describe efficiently the cognitive processes that underlie the student's actions along with the student's behavioral patterns and preferences.
- iii. Based on the aforementioned characteristics, the system creates clusters of the already existing students. These clusters contain valuable information about their members, considering their behavior, their preferences and generally their interaction with the system.

Finally, when a new user is added to the system, by registration, the student's initial characteristics (age, level of knowledge) are processed by the system's machine learning mechanisms to classify the student in an existing student cluster. In this way, it can offer personalized advice for revision end error proneness concerning the taught languages, individualized navigation and adaptation to student's needs.

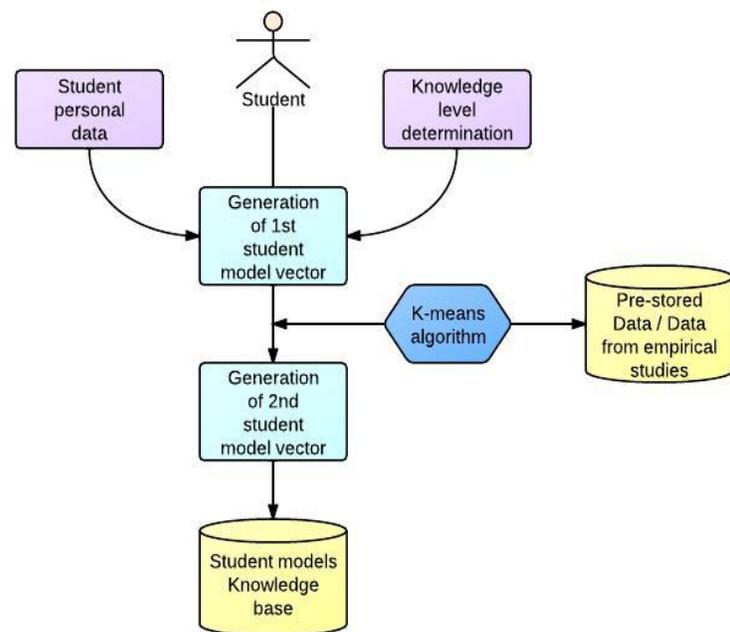


Figure 3. Architecture of the system

#### IV. OVERVIEW OF THE SYSTEM

The system's user interface is extensively described in [10] and [11]. However, in this paper we focus on illustrating the mechanisms that lead to the successful creation of user models, related to the domain of knowledge. The following figures illustrate snapshots of the k-means operation. Specifically, Fig. 4 illustrates the characteristics of users as vectors. In Fig. 5, the k-means algorithm determines the means of the vectors. In Fig. 6, the corresponding Voronoi diagram is created. Fig. 7, illustrates a report of k-means, the initial user data and the resulting k-mean vectors. K-means algorithm creates a graphical representation of the points, which are the characteristics of users, namely the age and his/her knowledge level, as mentioned before. Furthermore, the system has the possibility of calculating the k-means. In addition, it creates the Voronoi diagram, determined by distances to a specified family of objects (subsets) in the space. The system is given a finite set of points in the Euclidean plane. In this case, each site is simply a point and its corresponding Voronoi cell consisting of all points whose distance to the aforementioned point is not greater than their distance to any other site. Each such cell is obtained from the intersection of half-spaces and hence it is a convex polygon. The segments of the Voronoi diagram are all the points in the plane that are equidistant to the two nearest generators.

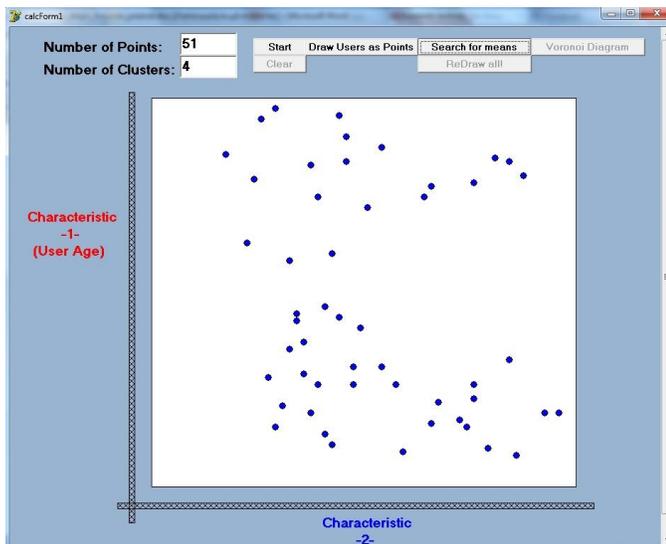


Figure 4. Illustration of users' characteristics as vectors

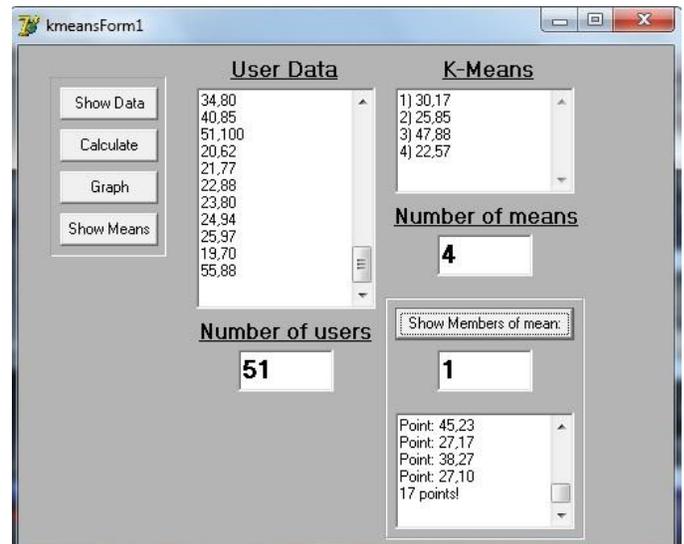


Figure 7. Report of k-means and user data

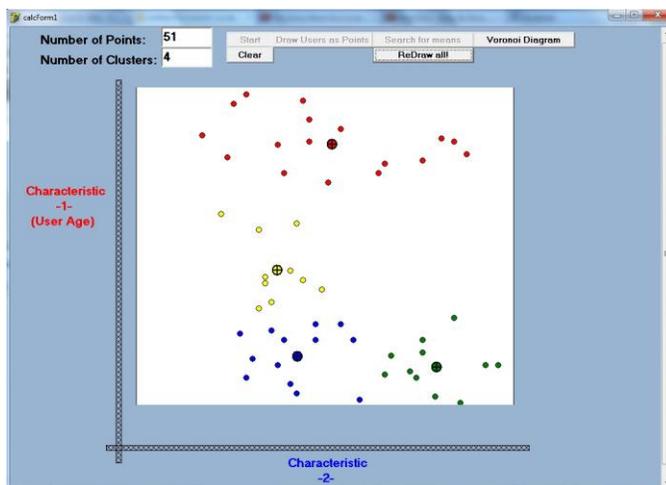


Figure 5. Determination of k-means

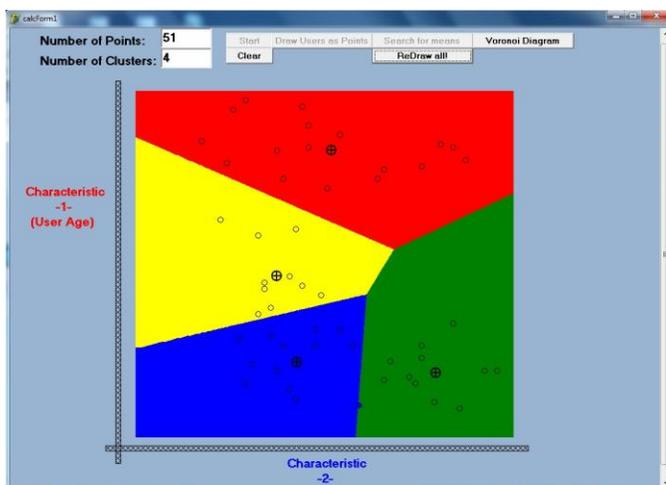


Figure 6. Corresponding Voronoi Diagram

## V. RESULTING SYSTEM

Our resulting a multiple language learning system, which offers the learning of three languages and offers communication between them, incorporates machine learning techniques for ameliorating the user modeling module.

The student's card, which is depicted in Figure 8, can be checked in every lesson of every language of our application. It constitutes a user model through which the student can monitor his performance in all the languages and whenever s/he wants.

Fig. 9 illustrates a categorization to a student's specific errors. The student can be evaluated and check where s/he is wrong and what type of mistake s/he has made. The different colors indicate different type of errors:

- i. The red color in the field means error in articles or pronouns.
- ii. The green color means a verb mistake.
- iii. The yellow color means a spelling mistake.
- iv. The blue color means confusion with the French language, while the purple means confusion with the English language
- v. Finally, the grey color indicates an unanswered question.

Fig. 10 illustrates a snapshot of the operating educational mobile application, where a student is completing a "fill in the gaps" exercise and taking the system's feedback. More specifically, in figure 10 we may see a student who has to fill in the gaps with the right word. The questions appear randomly and are adapted to each student model. It is quite important to note that the student must have acquired the knowledge offered by all the lessons as prerequisites.

The communication of the three languages of the system concerning the student's performance is quite noteworthy. Namely, the system can give advice to the user, as far as his/her performance in the other languages. This operation consists of a significant component of adaptivity, as the

system can store each student's performance in the database and can give advice, when it is necessary.

VI. CONCLUSIONS

In this paper, we have described an electronic learning platform that addresses the problem of grouping students in order to provide sophisticated user models. Clustering is conducted by the k-means algorithm which takes as input, to initialize the process, two important students' characteristics. Our approach to student modeling exploits the fact that educational systems have a large number of users, so in our approach we use a machine learning reasoning mechanism that recognizes similarities between users. The two key features of k-means that render this algorithm efficient is the fact that Euclidean distance is used as a metric and variance is used as a measure of cluster scatter and that the number of clusters k is an input parameter. Finally, after determining in which cluster each new student belongs, the system can reason about this specific student, adapting its behavior to the student's needs. The resulting adaptation emerges from the analysis of each cluster's characteristics that derive from each cluster as a superset of already existing student models.

It is in our future plans to evaluate our system in order to examine the degree of usefulness of the student modeling and error diagnosis components of our system. Moreover, we are planning to further evaluate the system in terms of its usefulness of the multilingual support along with the efficiency of k-means clustering in the language learning process.

REFERENCES

- [1] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney, A., "Auto tutor: An intelligent tutoring system with mixed-initiative dialogue", in IEEE Transactions on Education, vol. 48, pp. 612-618, 2005.
- [2] A. Niño, "Machine translation in foreign language learning: Language learners and tutors perceptions of its advantages and disadvantages", in ReCALL, vol. 21, pp. 241-258, 2009.
- [3] E. Christensen, P. Merrill, and S. Yanchar, "Second language vocabulary acquisition using a diglot reader or a computer-based drill and practice program", in Computer-Assisted Language Learning, vol. 20, pp. 67-77, 2007.
- [4] E. Friaiz-Martinez, S. Y. Chen, R. D. Macredie, and X. Liu, "The role of human factors in stereotyping behavior and perception of digital library users: a robust clustering approach", in User Modelling and User-Adapted Interaction, vol. 13, pp. 305-337, 2007.
- [5] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine Learning for User Modeling", in User Modelling and User-Adapted Interaction, vol. 11, pp. 19-29, 2001.
- [6] J. E. Beck, and B. Park Woolf, "High-level student Modeling with Machine Learning", in Proceedings of the 5th International Conference on Intelligent Tutoring Systems, Springer-Verlag, UK, pp. 584-593, 2000.
- [7] J. L. Plass, "Design and Evaluation of the User Interface of Foreign Language Multimedia Software: Acognitive Approach", in Language Learning & Technology, vol. 2, pp. 35-45.
- [8] K. Aytrk, O. Rahim, A. Keles, and A. Glc, "ZOSMAT: Web-based intelligent tutoring system for teaching-learning process", in Expert Systems with Applications, vol. 36, pp.1229-1239, 2009.
- [9] L. Amaral, and D. Meurers, "Conceptualizing student models for ICALL", in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4511 LNCS, pp. 340-344, 2007.
- [10] M. Virvou, and C. Troussas, "CAMELL: Towards a ubiquitous multilingual e-learning system", in CSEDU 2011 - Proceedings of the

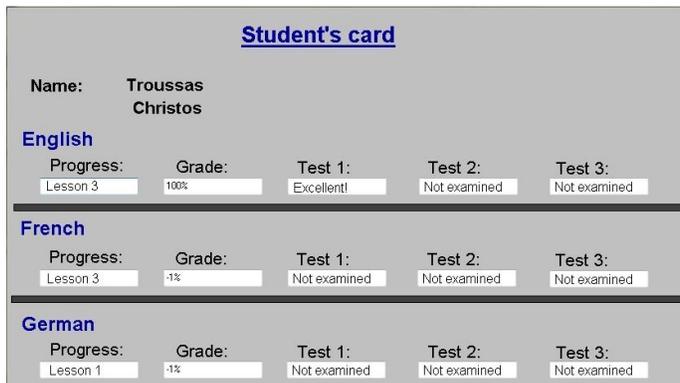


Figure 8. Student's profile

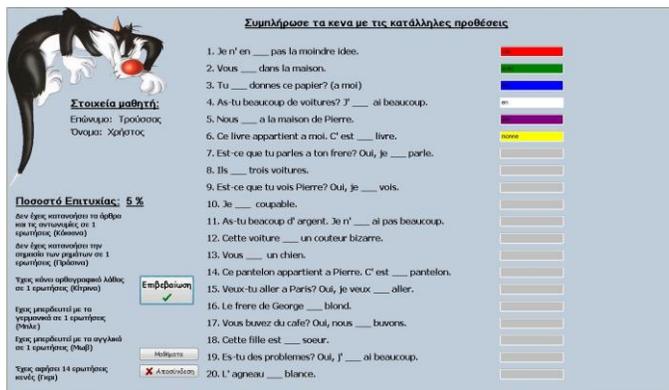


Figure 9. Error diagnosis process

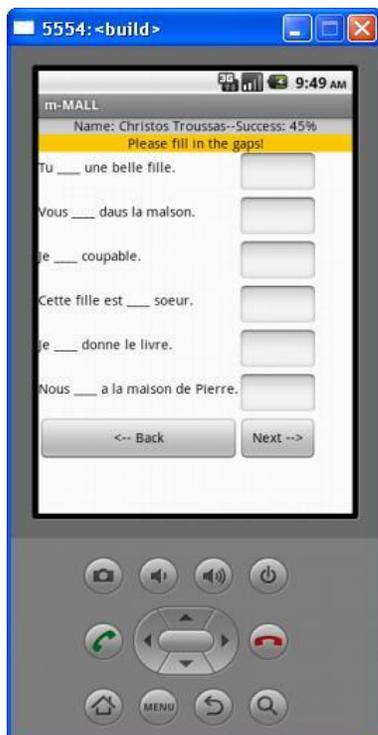


Figure 10. Fill-in-the-gaps test

- 3rd International Conference on Computer Supported Education, vol. 2, pp. 509-513, 2011.
- [11] M. Virvou, and C. Troussas, "Web-based student modeling for learning multiple languages", in International Conference on Information Society, i-Society 2011, Article number 5978484, pp. 423-428, 2011.
- [12] M. Virvou, D. Maras, and V. Tsiriga, "Student modelling in an intelligent tutoring system for the passive voice of english language", in Educational Technology and Society, pp. 139-150, 2000.
- [13] M. Virvou, E. Alepis, and C. Troussas, "MMALL: Multilingual Mobile-Assisted Language Learning", in Proceedings of the First International Symposium on Business Modeling and Software Design, pp. 129-135.
- [14] M. Virvou, and K. Chrysafiadi, "A web-based educational application for teaching of programming: Student modeling via stereotypes", in Proceedings - Sixth International Conference on Advanced Learning Technologies, ICAIT 2006, Volume 2006, pp. 117-119, 2006.
- [15] N. Ditcharoen, K. Naruedomkul, and N. Cercone, "SignMT: An alternative language learning tool", in Computers and Education, vol. 55, pp. 118-130, 2010.
- [16] N. Kurata, "Opportunities for Foreign Language Learning and Use within a Learner's Informal Social Networks", in Mind, Culture, and Activity, vol. 17, pp. 382-396, 2010.
- [17] O. Licchelli, T. M. A. Basile, N. Di Mauro, F. Esposito, G. Semeraro, and S. Ferilli., in Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), vol. 3029, pp. 935-944, 2004.
- [18] T. Basile, F. Esposito, and S. Ferilli, "Improving User Stereotypes through Machine Learning", in Communications in Computer and Information Science, vol. 249, pp. 38-48, 2011.
- [19] T. Finley, and T. Joachims, "Training structural SVMs when exact inference is intractable", in Proceedings of the 25th International Conference on Machine Learning, pp. 304-311, 2008.
- [20] V. Tsiriga, and M. Virvou, "A framework for the initialization of student models in web-based intelligent tutoring systems", in User Modelling and User-Adapted Interaction vol. 14, pp. 289-315.