

Personalized Teaching of Multiple Languages through the Web

Maria Virvou, Christos Troussas

Department of Informatics, University of Piraeus

Abstract

This paper describes the development of a personalized teaching system, which aims to be useful in multiple language learning. Web services provide remote access to the system, which performs affective user modeling. The system holds student models, namely it creates a profile for every student and provides individualized help, via the advice generator component. The error diagnosis component is also useful for evaluating students' errors, which is performed for all the system's supported languages. Given that software applications that operate over the web are targeted to a wide range of users, we promote the adaptivity and individualization to each user that interacts with the application and we give students the important facility of learning multiple languages over the Web. The student modeler may be authored by the instructor, along with all the functional components of our tutoring system.

1. Introduction

Major technological advances have led to the widespread use of the World Wide Web (WWW) and the Internet. WWW was developed to be a pool of human knowledge and human culture, which would allow collaborators in remote sites to share their ideas and all aspects of a common project. Hence, the development of web-based applications has become common place. Moreover, all the emerging needs of modern life accentuate the importance of learning foreign languages. Considering the scientific area of Intelligent Tutoring Systems (ITSs), there is an increasing interest in the use of computer-assisted foreign language instruction. The development of web-based educational systems has been a necessity. In this way, students may learn a foreign language, by using an application in the WWW. European reality necessitates the multiple language learning (www.europa.eu, [15]), so the students may further benefit from this educational process.

Due to the wide audience of language learning applications, there is a need to provide individualized interaction with users. In recent years, the development of personalized educational systems that operate over the WWW consisted of an increasing research effort. Adaptivity is an important issue in modern web-based educational systems that

aim at reaching a much more heterogeneous group of learners in settings where no teacher is available to help users during their learning process (Tsiriga and Virvou, [15]). The need for web-based tutoring systems that provide intelligence, interactivity and adaptivity is even greater when students are taught more than one foreign languages simultaneously. A solution to this problem may be the integration of the technology of ITSs with web-based instruction, so as to provide adaptive tutoring to individual students over the Web. Educational ITSs are software that aim at providing personalized instruction to students. ITSs offer intelligence and adaptivity to individual students' needs, via student modeling.

The classical architecture of an intelligent tutoring system is composed of four elements that are related: the Domain model, the Student Model, the Teaching Model and the User Interface (Virvou and Alepis, [16]; Antal and Koncz, [2]). The Domain Model handles all content to be taught by the system. The Student Model offers information about the behavior and the knowledge of the student and should be able to perform detection to the students' errors. The Teaching Model incorporates the knowledge of the system, consisting of a system of rules, which selects the content to be displayed and monitors the use of the student, and provides assistance when necessary. Finally, User Interface represents the model that interacts directly with the student.

In view of the above, we have implemented a web-based application, that is based on ITS's. Our system aims at providing cost-effective one-on-one tutoring. Like ITS's, we have tried to make it good at providing personalized instruction to students, because it 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 student modeling component includes modeling of students' knowledge and skills in the domain being taught, adaptation of instruction to his/her individual needs and declarative knowledge. 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 the related work, concerning web-based tutoring systems in section 2. In section 3, we discuss our system's architecture. Following, we discuss in

depth about the student modeling. As a next step, we present in detail the error diagnosis component along with the categorization of errors. In addition, we discuss the web-based education and multilingual support, while next we present the machine learning in student modeling. In section 4, we present a visualization of our system's basic process. Finally, in section 5, we come up with a discussion about the usability of web-based student modeling in multiple language learning and we present our next plans.

2. Related scientific work

ITSs are computer-based systems that provide individualized tutoring to students. Modeling students is used towards providing adaptivity in the educational process. The problem of implementing a student modeling component in a web-based environment is very significant and has attracted the interest from both educators and computer scientists.

Antal and Koncz [2] reviewed the student modeling problem for computer-based test systems and also proposed a novel method for the graphical representation of student knowledge. Chang and Hsu [4] introduce a computer-assisted language-learning system for use on PDAs, which integrates an instant translation mode, an instant translation annotation mode and an instant multi-users shared translation annotation function in order to support a synchronously intensive reading course in the normal classroom. Ditcharoen et al [7] presented a language learning tool for deaf people, which was designed to translate from Thai sign language into Thai text, by promoting the students' skills. Savvopoulos and Virvou [14] presented an intelligent tutoring component which helps elderly people in an adaptive way and predicts their mistakes. Moreover, Ferreira and Atkinson [9] presented a model of corrective feedback for an ITS for Spanish as a foreign language and proposed the design of a component of effective teaching strategies into this ITS. Dickinson et al [6] designed a paper-based system that provides feedback on particle usage for first-year Korean learners, who learn a second language. Furthermore, Katsionis and Virvou [12] presented a web-based educational game that performs affective user modeling by measuring emotional characteristics of users. Amaral et al [1] motivated a broader perspective of student models for Intelligent Computer-Assisted Language Learning that incorporates insights from current research on second language acquisition and language testing. Another computer-based program on second language acquisition is Diglot Reader, which was developed by Christensen et al [5] and 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.

Zhou et al [19] implemented a system which can give error trend relevant feedbacks to language learners through anatomy animation. Scooter the Tutor, developed by Baker et al [3], is a system which gives a gaming student supplementary exercises focused on exactly the material the student bypassed by gaming, and also expresses negative emotion to gaming students through an animated agent. AutoTutor is a system, developed by Graesser et al [10], which 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. Tsiriga and Virvou [15] presented a framework for the initialization of student models in web-based educational applications. Finally, Virvou et al [17] presented a novel approach for the evaluation of ITSs, which relies on an agent that may be used as a simulated student-user and incorporates modeling techniques. However, after a thorough investigation in the related scientific literature, we came up with the result that there are only web-based applications that deal with only one foreign language and few web-based applications which incorporate the student modeling component. Furthermore, there was no implementation of web-based educational applications, which promote the multilingual component along with individual and personalized student models for each student and perform error diagnosis to learners, through an intelligent way. The multilingual support enables students to learn two different languages in the same time and to accept advice concerning the learning material and the exercises.

3. Architecture of the system

The main role of our multilingual tutoring system is to play the role of the teacher in the educational process. For this reason, our system builds up an individual student model for every student in order to adapt the curriculum to him/her, to help him/her navigate through the course and to support working on exercises individually. The two main features of an ITS, that we incorporated in our system, are the curriculum sequencing and the interactive exercise answering support.

Curriculum sequencing describes the order in which students may learn new knowledge units along with the corresponding teaching operations, namely presenting examples and providing exercises and tests. Our web-based tutoring system provides guidance to learners so that they have specific idea concerning the ITS. For this reason, information that is contained in our individual student model can be used by the learning system to adapt the presentation of pages to the student. It is quite important to

distinguish between pages that contain new knowledge and are destined to be visited by the students and pages where learners have to perform successfully some tests in order to get evaluated and pass to the next lesson. In our web-based learning system, we maintain a student model in order to observe and diagnose the learner’s knowledge state.

Our system is destined to support students while they are working on multiple choice exercises and on test where they are asked to fill in the gaps. We use techniques to provide intelligent diagnosis of complete solution to exercises and continuous interactive exercise answering support during working at them. Our system observes the learner during the solving of an exercise in a specific language and gives advice concerning his/her performance in the other language. Student modeling is well suited for diagnosing complete and incomplete solutions to exercises and providing individualized help. The diagnosis follows the exercises answering procedure and is performed in the whole educational process.

The architecture of our system follows the main line of ITS architectures. It is agreed that the functional components of an ITS architecture are the domain knowledge, the student modeler, the advice generator and the user interface (Virvou and Alepis, [16]), as shown in Fig. 1.

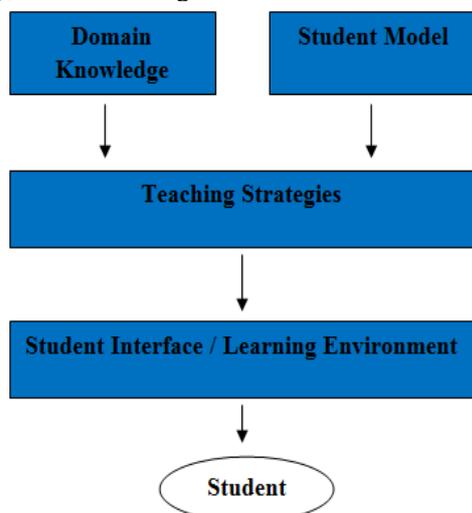


Figure 1. ITS architecture

The domain knowledge of our multilingual system consists 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.

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.

Our system’s multilingual component is quite significant, given that students may learn more than one languages in the same time through WWW. Furthermore, the student modeling component creates a profile for each student, which incorporates his/her performance and progress in each one of the languages, which are taught. The individualized error diagnosis process is performed by the student modeling component and is carried out through the two languages. Fig. 2 illustrates the architecture of the student modeling component in multiple language learning.

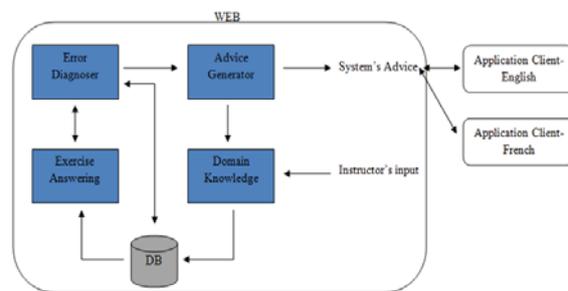


Figure 2. Architecture of our system

3.1. Student Modeling

Student modeling is the representation of the student's behavior and his/her state of knowledge. Student's behavior refers to a student's observable response to a particular stimulus (questions and exercises) in a given domain. The background knowledge of student modeling comprises:

- the concepts, principles and figures of a domain;
- the errors made by all the students in the same domain, namely a bug library.

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 (Virvou et al, [18]). 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.

Student modeling can undeniably benefit from the induction of knowledge which aims to render the students capable of solving exercises in a specific domain. A tutoring system may incorporate a student modeling component, which can use the system's bug library in order to model the students in an accurate and efficient way.

In conclusion, in this paper we try to rely on complete and consistent covering of examples in the multiple behavior set in order to induce a useful student model, which may help the multiple language learning, illustrated in Fig. 3.

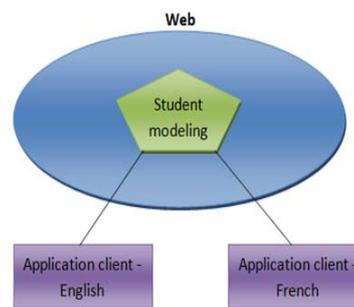


Figure 3. Student modeling component

3.2. Error Diagnosis

Our system is designed to promote noticing that will result in the improvement of students' existing grammatical knowledge. This can be accomplished by evaluating the students' performance through several tests.

Furthermore, our system includes two types of tests, namely the multiple choice exercises and exercises where the user is asked to fill in the gap inside a sentence.

Multiple choice questions are widely used, as they are a mainstay of achieving testing and also provided us with the ability to measure the students' achievements (Rodriguez, [13]). Multiple choice is a form of assessment in which the user is asked to select the best possible answer out of the choices from a list. One of them is the correct, while the others answers are erroneous. In our system, we have developed a bug library, which keeps all the erroneous answers and correlates them with a category of error, so that the student should have an integral idea of his knowledge and know exactly where s/he has weaknesses in.

In the test where the student is asked to fill in the gap inside a sentence, the system may examine the student's knowledge in all the lessons which have been taught in each language separately. There are twenty questions, which are different for each student and based on his/her user model. After completing the gaps, the system gives the results of the final examination. It shows the grade of the student and spots the erroneous answers. Furthermore, the system corresponds to the erroneous answers for each category of errors and gives a percentage that helps students understand in which lesson they have deficiencies.

Fig. 4 illustrates the relationship between the Student Modeling and Error Diagnosis components.

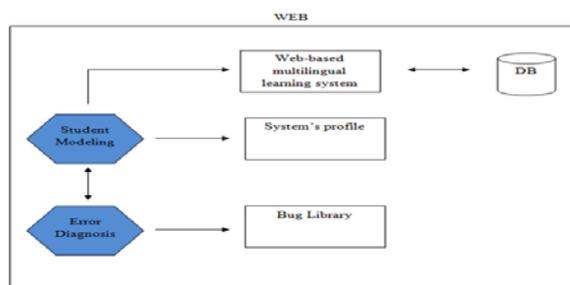


Figure 4. Student modeling and Error Diagnosis

3.3. Categorization of Errors

In the final test, the student has to fill in the gaps in some given sentences. The system recognizes the errors and associates them with the lessons taught by the system. The categories of error are:

1. Article and pronoun mistakes
For example the student may have used “a” instead of “the” or “she” instead of “they”.
2. Spelling mistakes
A spelling mistake is a result of letter redundancy, letter missing or interchange of two neighboring letters.
3. Verb mistakes
Verb mistakes occur when the user has typed another person than the correct one, for example s/he may have typed “he have” instead of “he has”.
4. 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.
5. Language Confusion
Our system is a multilingual web-based learning system, which means that a student may learn different languages at the same time. However, there is the possibility of student’s getting confused, concerning the proper use of an article or verb. Namely, the student may have used “I am” instead of “Je suis”, which is the French equivalent.

3.4. Web-based education and multilingual support

Using web services for the educational components can increase the interoperability and reusability of a web-based educational system and can greatly reduce the time and effort spent on building it. Our web-based ITS incorporates a student modeling component, which focuses on perceiving the users’ cognitive and emotional state while they answer the questions concerning the lessons taught.

Our system is an ITS with individual user models that is transferred over the Web. In order to accomplish this, we use a method for the deployment

of an ITS over the Web, namely we take all the parts of the client program and provide them through the Internet, in the form of an applet running on a Web page. In that way, the client-server program works by having the client as Web pages on the Internet and the server located at the Web server where the Web pages exist. The server is receiving information from the applet of the Web page. It is important that the client runs on a Web browser and is not installed on this client machine. We tried to pay more attention to knowledge sharing issues and standardization efforts in Web development.

By separating the student modeling and educational components, in the form of Web services, from the other parts of the ITS we can have these components work through the Internet while the other parts of the application are installed locally. Furthermore, these modeling components constitute the essential part of the reasoning of an ITS and might be reused by other educational applications. In this way, Web service implementations of ITSs can contribute to the reusability of their key functionalities if they are designed appropriately.

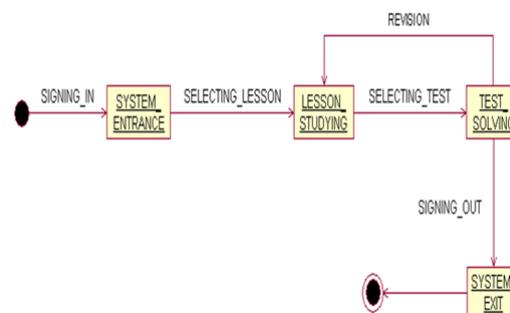


Figure 5. Activity Diagram

The use of student modeling can be closely related to web-based learning that can be shown by (Eklund, [8]):

1. The use of an expert’s construction of the domain to form the basis for the creation of links and nodes.
2. The incorporation of advanced navigational devices such as concept maps.
3. The providing of online help, which may be intelligent help if the student is modeled.
4. The use of adaptive interface based on stereotypical user classes to modify the environment to suit the needs of the individual learner.

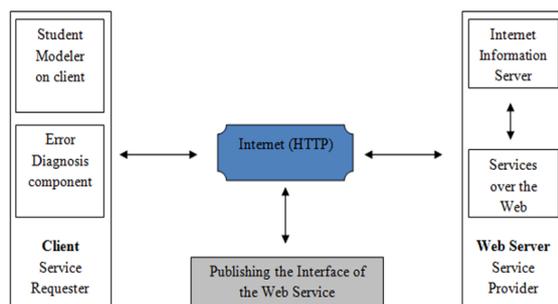


Figure 6. Web-based architecture

5. The providing of adaptive advice and model the student's acquisition of knowledge through their use of the environment (including navigational use, answers to questions, help requested), to intelligently suggest a preferred path through the knowledge base.

It is quite important to pay attention at the fact that the students are supported in the learning of two languages at the same time. Namely, the student may learn them over the web in an individualized and personalized way. Fig. 6 illustrates the process with which the web services operate.

4. Visualizing our system's blueprints

The following activity diagram in Unified Modeling Language (UML) is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel and whether there are alternative paths through the workflow. Using the activity diagram, we tried to describe the business and operational step-by-step workflows of components in our system. Namely, this activity diagram shows the overall flow of control. Fig. 5 illustrates the activities that can be done in parallel in our system.

5. Authoring process in our system

5.1. Authoring Domain Knowledge

Authoring domain knowledge comprises of creating new course structures as well as modifying the existing ones. The teachers have the permission to create exercises, define the right answer, erase an exercise or simply make several changes on them. Moreover, this module consists of setting up the repository. This process includes, inserting the available materials into the repository, as well as creating new learning materials. Given that our tool addresses to multiple language learning, we have implemented a platform where the teacher may edit the exercises of all the languages that are taught, in a simple way.

5.2. Authoring Student Model

Student Model is an abstract representation of the students' learning from the system, along with a student's profile, which includes personal information of him/her and other information concerning his/her performance and progress. It keeps track of student behaviors and makes inferences about what the student knows (Murray, 2003). A teacher can easily update a student's profile by interacting with the system, namely pressing buttons, choosing from a drop-down list and picking one from given multiple choices. Our system does not allow the teacher to make changes to the performance and the progress of the student. This prohibition ensures the integrity of the system. The student model can also be used as a "bug library", since misconceptions and buggy procedures can be defined to keep track of known classes of common deficiencies (Murray, 2003).

Furthermore, this module offers the possibility to the teachers to register a new student so that s/he is able to make use of the system and learn multiple languages. This registration can be conducted by filling in a simple form with the student's username, name and surname and gender.

5.3. Authoring of Teaching Model

This is an automated module, which works dynamically and makes pedagogic decisions, depending on the information of students' performance gathered at run time. Hence, the teachers do not have any direct control over the decision making procedures of this module. They get informed about the performance of the teachers, so that they can adapt the difficulty of the exercises to the level of each one student. This module promotes the individual and personalized learning and renders the students capable of filling all requirements of the educational procedure.

6. Case Studies

We are now going to present two case studies, one for the teacher and one for the student that use our learning system.

6.1. Case study for the Teacher

When the teacher is going to create multiple choice exercises, s/he can fill in the form with the sentence and then s/he can add four different choices and thereafter define the right choice. After the construction of a multiple choice exercise, the tool lets the teacher preview it and then s/he can add it into the system, by pressing a button of acceptance.

While students are tackling the given exercises, the system collects evidence about the level of difficulty so that it can provide feedback to the teacher. In this way, the teacher may edit a multiple choice exercise in order to adapt it to the level of knowledge of students. Nevertheless, there is one final test in the system that cannot be edited by the teachers. The students are asked to fill in the gaps of twenty given sentences. These sentences are chosen at random each time. The system may choose these sentences from the database, so that they are different for each student. In this way, we promote the diaphaneity in the students' performance evaluation.

The authoring tool also incorporates a user modeling component that conducts diagnosis to the students' errors. This mechanism is used by the system when a student tackles the final exercise and is explained in detail in the next section.

6.2. Case Study for the Student

The system recognizes each student by his/her user name and his/her password, which were defined upon his/her registration. The domain knowledge of the resulting system consists of three languages, which are English and French. 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 and in minuscule letters. 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 are 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, as they are a mainstay of achieving testing and also provided us with the ability to measure the students' achievements (Rodriguez, [13]). The multiple choice questions can be authored by the teacher at any time, as it was described above.

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-

exercises, then 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 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.

A matter of great importance is the existence of the final test, which is an exercise where the student is asked to fill in the gaps of a specific sentence. After completing the test, the student can check his/her performance and take advice about the point where the error lies and the categories of error. The questions of this test are adapted to the student's performance so they are different each time. While the student is in the process of answering this exercise, the system monitors his/her actions. If the student makes a mistake, the diagnostic component of the system will attempt to diagnose the cause of it. The diagnosis of the underlying cause of a mistake is a difficult task for an ITS because the observable symptoms need to be analyzed further. As Hollnagel [11] pointed out, there is an important distinction between the underlying cause or genotype of an error and the observable manifestation or phenotype of the error. In the final test, the system recognizes the errors and associates them with the lessons taught by the system. by recognizing the for-mentioned five categories of errors.

7. Conclusions

In this paper, we have described the crucial matter of student modeling in multilingual language learning over the Web. In addition, we discuss some basic issues on student modeling, such as:

1. Defining students' errors in the domain of knowledge.
2. Provide quick and didactic feedback.
3. Balance between the accuracy of student model and cost to build it.
4. Clarification of the inherent properties of the student modeling, as it contains domain-specific information.
5. Importance of primitives in error representation.
6. Inconsistency because of the change of students' understanding, the students' accidental slips and the students' inconsistent knowledge.

Furthermore, error diagnosis is performed dynamically for each student, taking into account information that originates from the student models. In order to perform error diagnosis, the system bears

a detailed categorization of common student's mistakes. The error diagnosis process of our system is especially focused on errors due to confusion of the other languages of the system, if the student learns more than on language at the same time. Apart from the friendliness of the user interface, our system is oriented to offer adaptivity and dynamic individualization to each user which interacts with the application, in order to promote the educational process by modeling the students who are taught multiple languages through WWW.

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 going to evaluate the usefulness of the multilingual support in web-based language learning process.

10. Acknowledgements

The authors would like to express their appreciation to Dr. Efthimios Alepis for his contribution and his useful comments.

11. References

- [1] Amaral, L., Meurers, D., "Conceptualizing student models for ICALL", in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Volume 4511 LNCS, 2007, pp. 340-344.
- [2] Antal, M., Koncz, S., "Student modeling for a web-based self-assessment system", in Expert Systems with Applications, Volume 38, Issue 6, 2011, pp. 6492-6497.
- [3] Baker, R.S.J.D, Corbett, A.T, Koedinger, K.R, Evenson, S, Roll, I, Wagner, A.Z, Naim, M, Raspat, J, Baker, D.J, Beck, J.E., "Adapting to when students game an intelligent tutoring system", in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Volume 4053 LNCS, 2006, pp. 392-401.
- [4] Chang, C.-K., Hsu, C.-K., "A mobile-assisted synchronously collaborative translation-annotation system for english as a foreign language reading comprehension", in Computer-Assisted Language Learning, Volume 24, Issue 2, 2011, pp. 155-180.
- [5] Christensen, E., Merrill, P., Yanchar, S., "Second language vocabulary acquisition using a diglot reader or a computer-based drill and practice program", in Computer Assisted Language Learning, Volume 20, Issue 1, 2007, pp. 67-77.
- [6] Dickinson, M, Eom, S., Kang, Y, Lee, C.M, Sachs, R, "A balancing act: How can intelligent computer-generated feedback be provided in learner-to-learner interactions?", in Computer Assisted Language Learning, Volume 21, Issue 4, 2008, pp. 369-382.
- [7] Ditcharoen, N, Naruedomkul, K, Cercone, N., "SignMT: An alternative language learning tool", in Computers and Education, Volume 55, Issue 1, 2010, pp. 118-130.
- [8] Eklund, J., "Cognitive models for structuring hypermedia and implications for learning from the world-wide web", 1995.
- [9] Ferreira, A., Atkinson, J., "Designing a feedback component of an intelligent tutoring system for foreign language", in Knowledge-Based Systems, Volume 22, Issue 7, pp. 496-501, 2009.
- [10] Graesser, A.C., Chipman, P., Haynes, B.C., Olney, A., "Auto tutor: An intelligent tutoring system with mixed-initiative dialogue", in IEEE Transactions on Education, Volume 48, Issue 4, 2005, pp. 612-618.
- [11] Hollnagel, E., "The Phenotype of Erroneous Actions", in International Journal of Man-Machine Studies, Volume 39, 1993, pp. 1-32.
- [12] Katsionis, G., Virvou, M., "Personalised e-learning through an educational virtual reality game using Web services", in Multimedia Tools and Applications, Volume 39, Issue 1, pp. 47-71, 2008.
- [13] Rodriguez, M.C., "Three options are optimal for multiple-choice items: A meta-analysis of 80 years of research", in Educational Measurement: Issues and Practice, pp. 3-13, 2005.
- [14] Savvopoulos, A., Virvou, M., "Tutoring the elderly on the use of recommending systems", in Campus-Wide Information Systems, Volume 27, Issue 3, pp. 162-172, 2010.
- [15] Tsiriga, V., Virvou, M., "A framework for the initialization of student models in web-based intelligent tutoring systems", in User Modelling and User-Adapted Interaction Volume 14, Issue 4, pp. 289-316, 2004.
- [16] Virvou, M., Alepis, E., "Mobile educational features in authoring tools for personalised tutoring", in Computers and Education, pp. 53-68, 2005.
- [17] Virvou, M., Manos, K., Katsionis, G., "An evaluation agent that simulates students' behaviour in intelligent tutoring systems", in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Volume 5, pp. 4872-4877, 2003.
- [18] Virvou, M, Maras, D, Tsiriga V., "Student modelling in an intelligent tutoring system for the passive voice of english language", in Educational Technology and Society, pp. 139-150, 2000.
- [19] Zhou, W., Zheng, J., Lu, Q., Chiu, T., You, X., Ye, W., "A computer assisted language learning system based on error trends grouping", in IEEE NLP-KE 2007 - Proceedings of International Conference on Natural Language Processing and Knowledge Engineering, Article number 4368040, pp. 256-261, 2007.
- [20] <http://europa.eu/>