User Modeling and User-Adapted Interaction **14:** 289–316, 2004.

A Framework for the Initialization of Student Models in Web-based Intelligent Tutoring Systems

VICTORIA TSIRIGA and MARIA VIRVOU
Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou St. Piraeus 18534, Greece. e-mail: {vtsir, mvirvou}@unipi.gr

(Received: 1 April 2003; accepted in final form: 13 September 2003)

Abstract. Initializing a student model for individualized tutoring in educational applications is a difficult task, since very little is known about a new student. On the other hand, fast and efficient initialization of the student model is necessary. Otherwise the tutoring system may lose its credibility in the first interactions with the student. In this paper we describe a framework for the initialization of student models in Web-based educational applications. The framework is called ISM. The basic idea of ISM is to set initial values for all aspects of student models using an innovative combination of stereotypes and the distance weighted k-nearest neighbor algorithm. In particular, a student is first assigned to a stereotype category concerning her/his knowledge level of the domain being taught. Then, the model of the new student is initialized by applying the distance weighted k-nearest neighbor algorithm among the students that belong to the same stereotype category with the new student. ISM has been applied in a language learning system, which has been used as a test-bed. The quality of the student models created using ISM has been evaluated in an experiment involving classroom students and their teachers. The results from this experiment showed that the initialization of student models was improved using the ISM framework.

Key words. initialization, machine learning for user modeling, stereotypes, student modeling, Web-based intelligent tutoring systems

1. Introduction

The widespread use of the WWW and the Internet has led to a trend towards the development of Web-based applications. Due to the diverse and wide audience of such applications, there is a need for them to provide more individualized interaction with users. Therefore, in recent years increasing research effort has been put into the development of personalized systems that operate over the WWW. This direction of research has also influenced the area of educational software. Adaptivity is a very crucial matter in 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. However, most existing Web-based educational applications lack the sophistication, interactivity and adaptivity of Intelligent Tutoring Systems (Weber and Specht, 1997).

A solution to this problem may be the integration of the technology of Intelligent Tutoring Systems (ITS) with Web-based instruction, to provide tutoring over the Web adaptive to individual students. ITSs are computer programs that aim at providing cost effective one-on-one tutoring. They are very good at providing personalized instruction to students, because they are designed to know who they teach, what they teach, and how to teach it. To a large extent in ITSs, intelligence and adaptivity are achieved by the incorporation of a student modeling component. The student modeling component attempts to model the student's knowledge and skills in the domain being taught and adapt instruction to her/his individual needs. Recently, several systems have been developed that make use of techniques from ITSs to provide individualized tutoring over the Web (Okazaki et al., 1996; Vassileva, 1997; Alpert et al., 1999; Heift and Nicholson, 2001).

Adaptivity can also be achieved in Web-based educational applications by the incorporation of techniques from the area of Adaptive Hypermedia. Adaptive hypermedia systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt the structure and content of the hypertext to the needs of that user (Brusilovsky, 1996). An increasing number of adaptive Web-based educational hypermedia systems have emerged during the last years (Weber and Specht, 1997; Brusilovsky and Pesin, 1998; Albrecht et al., 1999; Henze and Nejdl, 2001).

Adaptive hypermedia educational systems differ from ITSs in the sense that they use different techniques (e.g. adaptive presentation of the material and link adaptation) to personalize tutoring. These techniques are especially suitable for the construction of hypermedia electronic textbooks. ITSs on the other hand are very powerful at providing individualized support to students in their problem solving activity. A central component in the architectures of both Web-based ITSs and adaptive Web-based educational hypermedia systems is the student modeling component. Indeed, the student modeling module is the part of a Web-based educational application that is responsible for acquiring and representing the necessary information about each student. More specifically, the student modeling module performs two main functions (Nwana, 1991):

1. Initializes the student model when a new student logs on the ITS for the first time.
2. Updates the student model based on the student's interaction with the system.

Although a lot of research work has focused on the identification of efficient methods for updating the student model, the process of the initialization has often been neglected or it has been dealt using trivial techniques. When students start working with an ITS, the system has no prior knowledge about their proficiency level of the domain nor of their learning characteristics. However, the Web-based ITS attempts to provide individualized support. Therefore, the student modeler should have an efficient way of inferring initial information about the student. The initialization of a student model is of great importance because it seems unreasonable to assume that every student starts up with the same knowledge and misconceptions about the domain being taught. Indeed, an ITS runs the risk of losing its credibility and be considered as irritating and worthless to use by a student, if it fails to make plausible hypotheses about a student, before the student loses her/his patience with the system. Furthermore, misleading messages that do not correspond to the real strengths and weaknesses of the students may cause them frustration. Indeed, following bad advice may in many cases result in worse performance than getting no advice at all (De Bra, 2000).

In this paper we introduce a framework for the initialization of the student model in Web-based ITSs, which is called Initializing Student Models (ISM) framework. The ISM framework is a methodology that uses an innovative combination of stereotypes (Rich, 1979; 1983) and the distance weighted *k*-nearest neighbor (*k*-NN) algorithm (Dudani, 1976; MacLeod et al., 1987; Emde and Wettschereck, 1996) to set initial values for all aspects of the student model. In particular, a student is first assigned to a stereotype category on the basis of her/his knowledge level in the domain being taught. This is done based on the student's performance on a preliminary test posed to the student the first time s/he interacts with the Web-based educational system. Then, the distance weighted *k*-NN algorithm is used in order to initialize the model of the new student based on recognized similarities between the new student and other students who belong to the same knowledge level stereotype category. However, these students may have used the Web-based educational application for a period of time. In this case, the models of those students would have been individualized based on their actual behavior while interacting with the system. The similarity between students is calculated taking into account different student characteristics for different tutoring domains.

The ISM framework was implemented in a Web-based Intelligent Computer Assisted Language Learning (ICALL) System. Then we conducted an evaluation study, in order to assess the effectiveness of the ISM framework. According to the results of this evaluation, ISM was successful at providing sufficiently accurate initial student models, given the fact that very little is known about new students.

2. Related Work

2.1. APPROACHES FOR INITIALIZING THE STUDENT MODEL

Aïmeur et al. (2002) distinguish between three approaches for initializing the student model:

1. The ITS may assume that a new student knows nothing about the domain.
2. The student's prior knowledge may be evaluated by using a pre-test.
3. The system may use patterns among students in order to group similar students to categories.

For reasons of simplicity, a great number of educational systems initialize the models of new students by assuming that they know nothing or that they have some standard prior knowledge of the domain being taught. For example, da Silva et al. (1997; 1998) in their Web-based course for 'multimedia modeling and programming', assume at the beginning that a particular student has no prior knowledge concerning every concept in the domain knowledge and update the student's knowledge level of a certain concept only after s/he has visited the theory page related to this concept. Similarly, the student modeling framework described in (Tchétagni and Nkambou, 2002) does not initialize the model of a new student, but it infers the student's knowledge level only based on her/his interaction performance. An example of a system that initially assumes that a student who logs on the system for the first time has some standard prior knowledge in the domain being taught is the German Tutor (Heift and Nicholson, 2000; 2001). In particular, this system assigns new students to the 'intermediate' stereotype category concerning their knowledge level. Although this approach is the easiest way to address the problem of initializing the student model, it has poor performance for students who have different knowledge from the one initially assumed by the system.

The most direct way to initialize the model of a new student is by using exhaustive pre-tests that contain questions related to every topic in the domain being taught. This approach may be applicable in cases where the domain of interest is rather restricted. However, in case of a broader domain, using this method would require the student to answer questions for a long period of time before s/he could actually start working with the system. Indeed, users may be annoyed by being required to interact with a system and providing information without being aware of the use of this information. Furthermore, this time consuming process may disturb users due to the fact that it delays them from interacting with the system in a way that is meaningful for them (Schwab and Kobsa, 2002). An alternative that would reduce the number of questions needed to estimate the student's knowledge level would be the use of adaptive pre-testing. Adaptive pre-testing provides a dynamically generated, individualized test for each student. The decisions about the questions that will be included in the test are made while the student answers to the questions of the test. In particular, the choice of the next question that will be posed to the student is based on her/his answers to already posed questions (Guzman and Conejo, 2002). Using this approach, the system tries to draw inferences about the student's proficiency in a particular piece of knowledge (topic) of the domain being taught based on her/his answers to questions concerning another topic of the domain. Therefore, adaptive pre-tests should be very carefully designed taking into account the domain expertise (e.g. the relationships between the topics of the domain being taught). Furthermore, if the inferences drawn by the system are not correct for a particular student, or her/his answer to a certain question of the adaptive test is a simple guess or slip, then the accuracy of the student model is reduced. Examples of systems that use adaptive pre-tests for the initialization of the student model are MATHPERT (Beeson, 1989) and EDUCO (Kurhila et al., 2001).

An alternative approach for initializing user models is the stereotype approach. The stereotype approach was first introduced by Rich (1979) in a book recommendation system, called GRUNDY. Since then stereotypes have been used in many educational systems as a means for initializing the student model (Bontcheva, 2002; Garofalakis et al., 2002; Murphy and McTear, 1997; Virvou and Moundridou, 2001). Stereotype-based reasoning takes an initial impression of the user and uses this to build a detailed user model based on default assumptions (Kay, 2000). Although stereotypes are very powerful in providing considerable information based on few observations, they do not permit the formation of an accurate student model. The effectiveness of stereotype reasoning depends on the quality of the identified stereotypes, for example the number of different stereotypes supported by the system, the accuracy of the classification of users to stereotypes, and the quality of inferences that are drawn from stereotype membership (Kobsa et al., 2001). Therefore, before using such an approach, the developers of the ITS should conduct extensive empirical studies to ensure that the supported stereotypes are adequate. Furthermore, a problem of the stereotype approach to user model initialization is that it is quite inflexible due to the fact that stereotypes are constructed in a hand-crafted way before real users have interacted with the system and they are not updated until a human does so explicitly.

An interesting variation of the stereotypes approach is proposed by Aïmeur et al. (2002). They introduce CLARISSE, a machine learning tool for the initialization of student models. They have applied their initialization approach in an ITS for quantum information processing, named QUANTI (Aïmeur et al., 2001). CLARISSE, similarly with the work presented in Paliouras et al. (1999), uses a conceptual clustering approach in order to identify categories (clusters) in an initial set of students. Then, based on the identified categories of students, CLARISSE defines inclusion and/or exclusion rules for each cluster of students. Since the initialization of student models is performed based on the student's answers to an initial pre-test, these inclusion and exclusion rules refer to student answers in the questions of the pre-test. For example, an inclusion rule for some category could be that any student who gave a very inappropriate answer to a certain question would be included in this particular category. Therefore, students are classified in a category of students based on their performance on the pre-test.

Our approach to initializing the model of a new student shares some similarities with the stereotype approach and the methodology followed by CLARISSE. In particular, like CLARISSE, the ISM framework assigns students to stereotypes concerning their knowledge level in the domain being taught. However, instead of initializing the model of a student based on the default assumptions of the active stereotype, it uses information acquired from the models of other students who belong to the same stereotype. ISM makes use of the fact that these students have already interacted with the Web-based ITS sufficiently for the system to have been able to construct their models based on direct observations of their behavior. The initialization of the student model is performed based on the similarity of the new student with other students of the same stereotype category, with respect to certain domain-independent student characteristics that are important for each educational application. Furthermore, similarly with CLARISSE, the ISM framework makes use of a machine learning technique for the initialization of the student model. However, their approach differs from ours in the sense that they try to perform a higher level task, which is to learn groups of students based on their answers to a certain pre-test. In the case of the ISM framework, hand-crafted, predefined stereotypes are used in order to assign students to a certain category of students. Then, the model of a new student is initialized taking into account certain personal student characteristics, which may be selected by the designer, depending on the requirements posed by the particular tutoring domain of the application.

2.2. MACHINE LEARNING APPROACHES IN STUDENT MODELING

In this section we discuss how machine learning techniques have been applied in student modeling. The aim of this discussion is to highlight the similarities and differences of these approaches with our approach. We do not attempt to provide an exhaustive review of such student modeling approaches. A more detailed and informative review can be found in Sison and Shimura (1998). According to them, machine learning or machine learning-like techniques have so far been used in two areas of student modeling research:

1. to induce a single, consistent student model from multiple observed student behaviors, and
2. for the purpose of automatically extending or constructing from scratch the bug library of student modelers.

In the first case, the student modeler records the set of behaviors (which may contradict one another) that a particular student shows while interacting with the educational system. These behaviors serve as input data to a machine learning algorithm that is aimed at constructing a consistent description of the model of this student. Examples of systems that follow the first approach are DEBUGGY (Burton, 1982) and ML-Modeler (Gürer et al., 1995). In the second approach, the student modeler uses a machine learning technique in order to extend its background knowledge (in most of the cases its bug library). In this case, the input data to the machine learning algorithm is the set of behaviors of all the students that interact with the system. Then, based on this information, the student modeler infers new background knowledge (e.g. bugs) that has not been predefined. The newly acquired knowledge is used by the system in subsequent interactions of students. Systems that have followed such an approach include PIXIE (Sleeman, 1987), the system presented in (Hoppe, 1994) and MEDD (Sison et al., 1998; 2000). Furthermore, ASSERT (Baffes and Mooney, 1996) uses machine learning both for the construction of a consistent student model and for the dynamic creation of the student modeler's bug library.

More recently, a number of systems have also used machine learning techniques in order to make inferences concerning higher level information about students. For example, Chiu and Webb (1998) try to predict future actions of a particular student by using the Feature Based Modeling technique, whereas Beck and Woolf

(2000) have used Linear Regression to determine how likely a student is to answer a problem correctly and how long it will take her/him to generate this response. Moriarty et al. (2001), on the other hand, have developed a system for providing students with personalized multiple choice exams. The system is able to predict student performance in a particular exam, using the k-NN learning approach. In particular, the system tries to predict a student's answer to a specific multiple choice question based on other student's answers to this question. The students considered are those that are highly 'close' to the student in question. The closeness of other students with the student in question is specified in terms of the similarity of other students' answers to questions with the answers of the student in question.

In the case of the ISM framework, we have used a machine learning algorithm in order to address a totally different problem, which is the initialization of the model of a new student. Our approach has some similarities with the second category of systems in terms of the classification mentioned above. In particular, we also use information held in the models of all students interacting with the Web-based ITS in order to perform a task. However, the reason for using this information in the ISM framework is totally different. In our case, the models of other students who have been registered to the educational system serve as indicators of the initial knowledge level and difficulties of the new student. Furthermore, the task of setting initial values to the model of a student in ISM is not concerned with higher level characteristics of the student, such as the prediction of the amount of time a student will spend in a test. In ISM, machine learning is used as a means to assess the prior knowledge level and the error proneness of the student in each concept of the domain being taught.

3. Architecture of the ISM Framework

In this section we describe the architecture of the ISM framework (Figure 1). In particular, according to the ISM framework, initial information about a new student is acquired by the Web-based educational system, based on an interview and a preliminary test. At first, the student is interviewed about personal characteristics that are required for the student model. The interview takes place the first time that a student interacts with the system. It contains questions related to certain personal, domain independent data, such as the student's name, age, etc. as well as several indirectly domain dependent characteristics. For example in the case of an educational application for language learning, the indirectly domain dependent characteristics include the mother tongue of the student.

Asking students to provide information about themselves is one of the most direct methods of acquiring information. However, asking students about themselves is not always the best method. For example, self-assessment is error-prone, since users are often not correctly aware of their own capabilities (Hothi and Hall, 1998; Kobsa et al., 2001). For the above reason, according to the ISM framework, a preliminary test should be used in order to assess the knowledge level of the student concerning

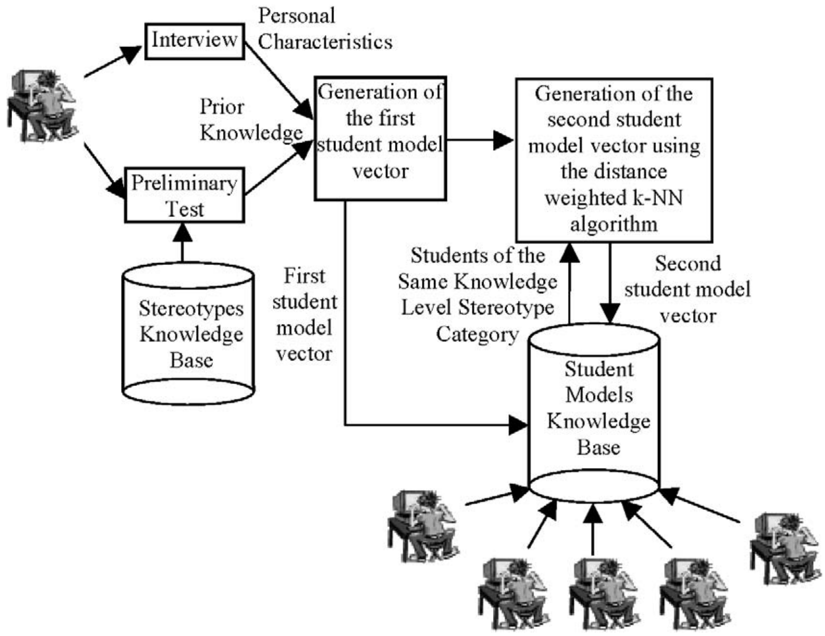

Figure 1. Architecture of the ISM framework.

the domain being taught and/or certain important prerequisite topics. In particular, the preliminary test should be designed so as to contain representative questions that cover the whole domain being taught and also if necessary the important topics that should be known prior to studying the domain of interest. According to the student's performance on the questions of the preliminary test, the Web-based educational system assigns the student to a stereotype category concerning her/his knowledge level.

According to Kay (2000), a stereotype consists of three main components,

(a) a set of trigger conditions,
(b) a set of retraction conditions, and
(c) a set of stereotype inferences.

The trigger conditions are boolean expressions that activate a specific stereotype, whereas the retraction conditions are responsible for deactivating an active stereotype. Furthermore, once the user is assigned to a stereotype, the stereotype inferences of this particular stereotype serve as default assumptions for the user.

According to the ISM framework, the Web-based ITS should classify students into stereotypes concerning their prior knowledge level in the domain being taught. In particular, the stereotype that a new student belongs to is used as an attribute that defines the number of students that will be considered as the nearest neighbors and will be taken into account for the initialization of the model of the new student. Furthermore, the system should be able to alter the stereotype that is active for a specific student that receives instruction by the system and therefore her / his knowledge may evolve. If this was not done, the system would falsely initialize the model of a new student using information about other students who have been interacting with the system for a length of time and have become more proficient in the domain.

In the ISM framework, the default assumptions of each stereotype are not always used as such, but are refined by taking into account the actual behavior of the other students that belong to this stereotype. The contribution of these students to the initialization of the new student model is weighted based on their similarity with the new student. Thus, the default assumptions of each stereotype are only used if there are no other students known to the system and belonging to the same stereotype as a new student. In such cases, the only information that can be used for a new student comes from the default assumptions of the active stereotype. Otherwise, the system assumes that the new student has similar behavior as the observed behavior of the known students that belong to the same stereotype. The models of those similar students may have been individualized based on their actions when they interact with the system. However, according to the ISM framework, all students of the same stereotype participate in the initialization process, irrespective of the amount of time they have spent interacting with the Web-based ITS.

The framework does not limit the number of the supported knowledge level stereotypes. However, the decision concerning the number of stereotypes affects the number of similar students that will be used to make inferences concerning the new student. The larger the number of different stereotypes, the smaller the number of nearest neighbors that will participate in the classification task.

The classification of students to stereotypes is based on the students' actions as well as the system's knowledge of the domain. This is done using the trigger and retraction conditions of the stereotypes. In particular, the trigger conditions of the stereotypes concern the student's performance on a preliminary test as well as the student's mastery of the domain concepts as they receive instruction by the system. For example, in the case of the preliminary test, a trigger condition may be associated with the number of questions that have been answered correctly. Furthermore, if the system has sufficient information concerning the student's mastery of the domain concepts based on direct observations, a trigger condition may be the student's mastery of some concepts that are considered 'mundane'. The retraction conditions are used in order to alter the active stereotype if the student is observed to have different behavior from that expected by the active stereotype. For example, if the student is observed to make many mistakes in concepts that the system considers simple, though the active stereotype assumes that s / he has mastered all simple concepts, then a retraction condition should be used to alter the active stereotype for this student. Furthermore, if a student is observed to have mastered all simple concepts due to the instructions s / he has received by the system, then a retraction condition should be used to deactivate the knowledge level stereotype of the student. In this case, a triggering condition of a more advanced stereotype will activate this stereotype for the student. In our approach, the stereotype inferences concern default assumptions about the student's mastery of the domain knowledge concepts and the student's proneness to make mistakes in these concepts. These default assumptions should be specified based on empirical studies that involve domain experts, human teachers and students.

The information acquired from both the interview and the preliminary test is represented as a feature vector, which is of the form:

$$\langle \text{Student\_Code, Name, Stereotype, Characteristic}_1, \text{Characteristic}_2, ..., \text{Characteristic}_n \rangle$$

The n Characteristics of this vector record the student attributes that may have an effect in the student learning of the domain being taught by the ITS. These characteristics have to be specified prior to the development of the Web-based ITS. In particular, the specification of these important student attributes may result either from teachers alone or after the conduction of an empirical study that involves domain experts, human teachers and students. For example, in the case of a language tutor, the first student model vector may be the following:

⟨Student_Code, Name, Stereotype, Carefulness, Mother_Tongue, Chinese, English, Finish, French, German, Greek, Italian, Russian, Spanish, Turkish⟩

This vector contains information that concerns the name of the student, the knowledge level stereotype that the student belongs to, her/his degree of carefulness, her/his mother tongue as well as the other languages that the student already knows. Among those characteristics, the knowledge level of the student may be inferred based on the students' performance on the preliminary test whereas the other characteristics may be obtained by the interview presented to students. These student attributes, may then be used by the language tutor to define the similarity between students.

The initial information that has been acquired directly from the student, as well as information from existing students is then used in order to produce a second vector that represents the system's estimations of certain domain dependent attributes of the new student. The second student model vector is of the form:

$$\langle \text{Student\_Code, Domain\_Related\_Characteristic}_1, \text{Domain\_Related\_Characteristic}_2,...,\text{Domain\_Related\_Characteristic}_m \rangle$$

At this point, the m Domain_Related_Characteristics can be associated with student attributes that take values from a continuous set of real values. For example one such attribute could be the degree of knowledge of the student for each concept in the domain knowledge and another could be the error proneness of the student in each domain concept. In the case of the above described language tutoring system, the second student model vector may be the following:

⟨Student_Code, (Know_Concept1, Errors_Concept1), (Know_Concept2, Errors_Concept2),...⟩

This vector is initialized using the distance weighted k-NN algorithm and it contains information concerning the student's knowledge level and error proneness in each concept of the domain knowledge of the language tutor.

The first student model vector serves as the input to the distance weighted k-NN algorithm so that the initialization task is performed. In our approach to produce the second domain related feature vector, we take weighted sums of known values to produce a value for an unknown quantity. In particular, for each unknown estimation of a particular student characteristic, the known values are the estimations of this characteristic, acquired from the models of other students that belong to the same knowledge level stereotype category with the student in question. The weights are a measure of the similarity between the student in question and the other students of the stereotype category. However, in cases where there are no other students that belong to the same stereotype category with the new student, the initialization of the model of this student is based on the default assumptions of the stereotype that has become active for the student.

For example, let us assume that the above mentioned language tutoring system already has the first student model vectors that are presented in the first four lines of Table I in its student models knowledge base. If a new student, after completing the interview and the preliminary test was found to be described by the first student model vector that is presented in the last line of Table I, then the system would initialize her/his student model using information only from students with codes 'Stu_1' and 'Stu_4'. This is due to the fact that these two are the only students that belong to the same stereotype (beginner) with the new student. Furthermore, when estimating the degree of knowledge and the error proneness of the new student, the information of the second student model vector of 'Stu_1' would have a greater contribution. This is due to the fact that 'Stu_1' is more similar to the new student as compared to 'Stu_4'.

In view of the above, the ISM framework aims at refining the inferences drawn from the classification of a new student to a stereotype concerning her/his knowledge level. This is achieved by consulting the models of other students who belong to the same knowledge level stereotype and share some similarities with the new student. These similarities concern certain student characteristics that can influence the way students learn the domain being taught by the Web-based ITS. In this way, the student modeler has the capability of dynamically refining the default assumptions of a particular stereotype based on the actual observed behavior of students that belong to this stereotype. These refined assumptions are then used for the initialization of the models of new students who are classified to the stereotype.

4. Distance Weighted k-NN Algorithm in ISM

The distance weighted k-NN algorithm is a refinement of the original k-NN algorithm (Cover and Hart, 1967; Dasarathy, 1991). In general, nearest neighbor

Table I. First student model vectors of a language tutoring system

Student code	Name	Stereotype	Degree of care-fulness	Mother tongue	Chinese	English	Finish	French	German	Greek	Italian	Russian	Spanish	Turkish
Stu_1	Jim	beginner	Careful	Greek	no	no	no	yes	yes	yes	no	no	no	no
Stu_2	Sofia	intermediate	Careful	Greek	no	no	no	no	no	yes	yes	yes	no	no
Stu_3	Mary	novice	Careless	Russian	no	no	no	yes	no	no	no	yes	no	no
Stu_4	Panagiotis	beginner	Careless	Spanish	no	no	no	yes	no	no	yes	no	yes	no
Stu_5	Alex	beginner	Careful	Greek	no	no	no	yes	no	yes	no	no	no	no

learning algorithms typically store all of the *n* available training examples during learning. These algorithms use a distance function to determine how close a new query instance is to each stored instance, and use the nearest instance or instances to classify the query instance (Wilson and Martinez, 1997). The basic idea of the distance weighted *k*-NN algorithm is to weigh the contribution of each of the k neighbors according to their distance to the query point, giving greater weight to closer neighbors (Mitchell, 1997).

When applying a distance weighted *k*-NN algorithm, the main decisions that have to be made are the following:

1. Which function will be used to measure the distance between students and which features would formulate the input space of the distance function?
2. How many neighbors (*k*) will participate in the classification task and which function will be used to classify new instances?

In the subsequent sections, we describe the approach taken by the ISM framework to address the above issues.

4.1. MEASURING THE SIMILARITY/DISTANCE BETWEEN STUDENTS

In different application domains, different student characteristics may be considered important for deciding how close a student is to another. For example, in an ICALL system, a student attribute that may be of great importance could be the mother tongue of the student, whereas, in an algebra tutor, the student's performance on simple arithmetic operations (addition, subtraction, multiplication and division) could be significant for defining the similarity between two students. Therefore, the first task in the application of ISM is the specification of the characteristics that will formulate the input space of the distance weighted *k*-NN algorithm. The involvement of experts of the domain being taught and human teachers is very important in this process, since they are the most appropriate source for providing such information. As stated previously, the values for all the important characteristics that will be used to classify a new student have to be acquired when the student registers to the Web-based ITS and should be represented using a feature vector.

In order to accommodate all possible types of values of the characteristics used to calculate the distance between students, ISM uses a heterogeneous distance function that can handle both nominal (e.g. the mother tongue of a student) and real values (e.g. the student's percentage of correct answers to questions of a preliminary test that concern simple arithmetic operations). Thus, ISM uses a distance metric similar to the one used in IB1, IB2, and IB3, which are systems described in Aha et al. (1991). In particular, the standard Euclidean distance is used in order to calculate the distance between two real valued attributes and a simple overlap metric for nominal values. Furthermore, if either of the attribute values is unknown, the distance metric assumes that the distance between these two values is the maximum possible. Hence, the distance between two values x and y of a given attribute a is computed using the following formula:

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ is unknown} \\ \text{overlap }(x,y), & \text{if } x,y \text{ are nominal values} \\ \sqrt{(x-y)^2}, & \text{if } x,y \text{ are real values} \end{cases} \tag{1}$$

where

$$\text{overlap}(x,y) = \begin{cases} 0, & \text{if } x = y \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

One problem with the overlap metric used in ISM to calculate the distance between nominal values is that it treats all different values in a similar way irrespective of how close they may be. For example, if this metric was used to define the distance between two values that concerned the degree of carefulness of students, it would consider as equally different the case of a careful and a careless student and the case where the students are careful and moderately careful respectively. In a future version of the framework, we intend to refine the overlap metric so as to be able to differentiate between ranges of nominal values.

Having defined a way to calculate the distance between two values of a given attribute, we should now define a function that calculates the overall difference measure of two students. In ISM, the overall distance between two students s_a and s_b is calculated as:

$$\Delta(s_a, s_b) = \sum_{a=1}^{n} d_a(x,y) \tag{3}$$

where n is the number of attributes that are used to measure the distance between two students.

4.2. CLASSIFICATION FUNCTION

The main process of the algorithm is the classification of an object based on the feature vector of this object and the feature vector of the k neighbors that are near this object. One important aspect of this process is the definition of the number of neighbors that will participate in the classification task. In ISM, the number of neighbors (k) is set to be the number of students that belong to the same stereotype category with the student in question. This is due to the fact that students who belong to different stereotypes are not expected to have similar knowledge of the domain, irrespective of the other characteristics that play a significant role in the learning process. For example, intermediate and advanced students may have similar knowledge concerning some simple concepts. However, they are not expected to have similar knowledge when it comes to concepts that are considered complex in the domain.

Furthermore, a classification function has to be specified that uses as input the instance that should be classified (new student) and the k nearest neighbors (students that belong to the same stereotype). In ISM, the system sets initial values to the estimations of the student's degree of knowledge and error proneness for each concept in the domain based on the known values of these attributes that are acquired by students that have already been registered to the Web-based ITS. In order to predict the values of the degree of knowledge and error proneness of the new student (s_q), we use a distance weighted mean value of the degree of knowledge and error proneness of the k students that belong to the same stereotype with the new student ($s_1, s_2, \ldots, s_k$). The vote of each of the neighboring students is weighted according to the inverse square of its distance from s_q. Therefore, for each concept in the domain knowledge (Concept$_x$), the function that estimates the degree of knowledge of the new student (s_q) is calculated using the following formula:

$$\text{Knowledge\_Level}(\text{Concept}_x, s_q) = \frac{\sum_{i=1}^{k} w_i \ \text{Knowledge\_Level}(\text{Concept}_x, s_i)}{\sum_{i=1}^{k} w_i} \quad (4)$$

where w_i is the weight of the contribution of each student and is calculated as:

$$w_i = \frac{1}{\Delta(s_q, s_i)^2} \quad (5)$$

The error proneness of the new student concerning a concept (Error_proneness (Concept$_x$, s_q)) is estimated in a completely similar way, as shown the following formula:

$$\text{Error\_Proneness}(\text{Concept}_x, s_q) = \frac{\sum_{i=1}^{k} w_i \ \text{Error\_Proneness}(\text{Concept}_x, s_i)}{\sum_{i=1}^{k} w_i} \quad (6)$$

To accommodate the case where the query student s_q matches exactly one of the students (s_i) that is used as a training example and the denominator $\Delta(s_q, s_i)$, is therefore zero, we assign w_i to be equal to 1 (maximum weight) in this case. Furthermore, in case there are no students belonging to the same stereotype with the new student, then her/his student model is initialized using the default assumptions of the active stereotype.

5. Application of the Approach to a Web-based ICALL

The first system that used ISM for initializing the model of a new student was Web-Passive Voice Tutor (Tsiriga and Virvou, 2002a). Web-Passive Voice Tutor (Web-PVT) is an adaptive and intelligent Web-based tutoring system that aims at teaching non-native speakers the domain of the passive voice of the English language. The early versions of Web-PVT did not have a sophisticated method for the initialization of student models (Virvou and Tsiriga, 2001). In the case of Web-PVT the ISM framework is instantiated by assuming that students will have similar strengths and weaknesses when they learn English if they begin with a similar knowledge level of English, if they have the same mother tongue, know the same foreign languages and have the same degree of carefulness when solving exercises.

The current version of the system constructs an individual model for each student that contains information about the knowledge level and the error proneness of the student in each concept of the domain knowledge. The system then uses this model to support the student while studying theory and solving exercises. In particular, based on the information that concerns the knowledge level of the student in each concept of the domain knowledge, the system provides individualized support when s/he navigates through the course material. Web-PVT uses a combination of two link adaptation techniques to help the student while navigating through the structured theory hyperdocument; namely adaptive link annotation and direct guidance (Brusilovsky, 1996). Furthermore, the system uses information on the student's knowledge level of concepts in order to select an exercise for the student to solve. The error proneness of the student, on the other hand, is used for error diagnosis. More specifically, this piece of information is used by Web-PVT in cases where the system has to disambiguate between competing hypotheses that concern the cause of students' mistakes. For example, if a student is given the sentence: 'Mary gave me a gift' and is asked to convert it to passive voice, the correct answer would be 'I was given a gift by Mary'. However, if the student types the sentence 'I was give a gift by Mary' where the verb 'give' is not in the past participle, then this mistake may be attributed to one of two categories of error. It could either be an accidental slip, caused by the student's carelessness or a mistake concerning the concept of 'verb tense conversion'. If this particular student has low error rates for the concept of 'verb tense conversion' and furthermore s/he was considered careless, then the system would favour the accidental slip as the most probable cause of the ambiguous mistake for this student.

5.1. ACQUIRING AND REPRESENTING INITIAL INFORMATION ABOUT THE STUDENT

Web-PVT acquires initial information about a new student when s/he interacts with the system for the first time. The student is expected to provide personal information by answering simple questions in an initial interview (Figure 2). The first five questions of the interview concern the student's record. The following two questions are related to the student's mother tongue and prior knowledge of other languages. Finally, in the last question the student is asked to give a self-estimation of how careful s/he is while solving exercises. In particular, the student can select among three different categories, namely careless, averagely careful, and careful. Indeed, this feature is considered important for finding similarities among students. This is so because many students are quite anxious to answer questions quickly in tests and they do it in a hasty manner that results in errors. However, errors due to lack of carefulness do not mean lack of knowledge. Therefore, it is important for the system to know the difference when it makes error diagnosis. Moreover, students are considered capable of assessing their degree of carefulness themselves. This

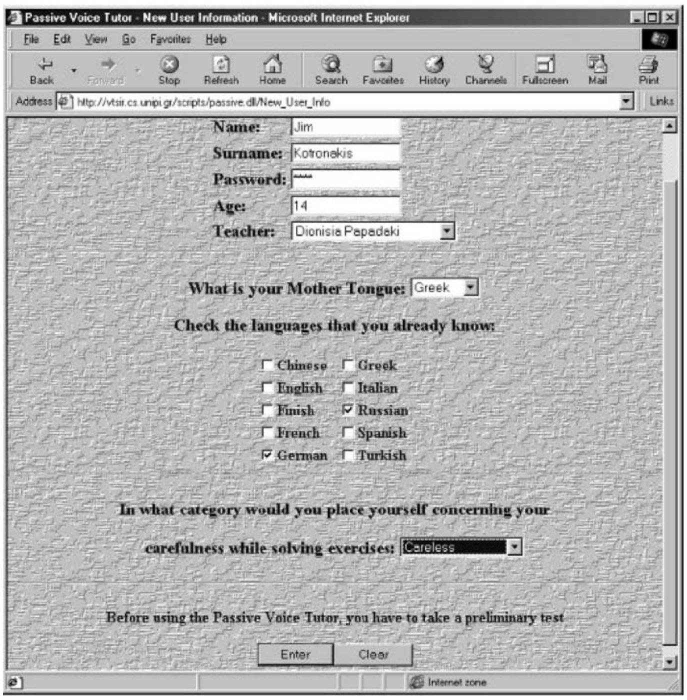

Figure 2. Initial interview in Web-PVT.

is so, because carelessness or carefulness is a domain-independent feature that a student has when responding to questions in all domains. Therefore, they are expected to have had feedback from many instructors and tutors of various domains on this feature from courses they had attended prior to this one. This feedback must have given them an idea of how careful they are in general.

Furthermore, Web-PVT also uses a preliminary test in order to assess the initial knowledge level of the student concerning the passive voice of the English language. The test has been constructed by human experts so as to contain representative questions that cover the whole domain of the passive voice of the English language. The preliminary test is given to students before they have ever interacted with Web-PVT. Then, based on the student's performance on the preliminary test, s/he is classified into one of the four distinct stereotypes, namely novice, beginner, intermediate and advanced. The definition of the stereotypes was based on an empirical study that involved teachers of English and their students. This study was conducted over a period of two months and it resulted in the identification of the triggering and retraction conditions, as well as the inference rules of the four stereotypes. In Web-PVT, stereotype inferences are default assumptions that concern the knowledge level of the student and her/his proneness to make mistakes in each concept of the domain, based on the difficulty level of the concept. The concepts may be associated with one of three levels of difficulty, namely simple, mundane, complex. An example of an inference rule in the 'intermediate' stereotype is that her/his knowledge level in all simple concepts is very high and that s/he never makes mistakes when using these concepts in exercises.

When the necessary information about the student has been acquired, the system needs to properly represent the student characteristics so that they could be further exploited. According to the ISM framework, the student model is represented as a set of feature vectors. The first vector is responsible for representing information acquired by the student in her/his first interaction with the system. In Web-PVT, the characteristics contained in the first vector include the name of the student, the stereotype category that s/he belongs to, an estimation of how careful the student is while solving exercises, her/his mother tongue, as well as other languages that the student already knows.

The choice of the student characteristics that formulate the first vector is based on the fact that students' performance in language learning is greatly influenced by the issue of language transfer. Language transfer is the interference resulting from the similarities and differences between the target language and any other language that has been previously (and perhaps imperfectly) acquired (Ogata et al., 2001). Indeed, the kinds of error a student makes is greatly influenced by the mother tongue of the student and/or foreign languages s/he may be learning. Furthermore, these characteristics play a role in the difficulty a student faces in acquiring a new piece of knowledge. For example, students who have French as their mother tongue should not have difficulty in understanding the grammatical piece of knowledge that concerns the passive voice of the English language. This is so because in French, the passive voice form is used in similar ways to English. However, the acquisition of this grammatical form is not equally easy for students who have a mother tongue where the passive voice is not used so much.

Furthermore, the proficiency level of the student in the domain being taught also plays a significant role in the student's proneness to make mistakes of a particular type, irrespective of the native language of the student and/or the foreign languages s/he may be learning. Indeed, intermediate students who have Greek as their mother tongue may type 'the police has arrested him' instead of 'the police have arrested him' due to language transfer. However, in case of an advanced student, this error is not expected with the same frequency. Finally, the degree of carefulness of a student may be a way to explain a certain category of mistakes. For example, if a student who is considered 'advanced' concerning her/his knowledge level types the sentence 'Expensive cars are *drive* by John,' then the most probable cause of the mistake would be the student's carelessness. This is due to the fact that it seems unreasonable to assume that an advanced student does not know how the simple present tense is transformed to the passive voice. In view of the above, the first student model vector in Web-PVT is defined as presented in Section 3.

5.2. SETTING INITIAL VALUES TO THE DOMAIN-RELATED VECTOR

The aim of the initialization process according to the ISM framework is to produce a domain related vector that represents the system's estimations about the degree of knowledge and error proneness of the student for each concept in the domain knowledge. In particular, for each one of the 42 concepts that are contained in the domain knowledge of Web-PVT there are two feature-value pairs related to it in the student model. The first pair represents an estimation of the student's degree of knowledge concerning this particular concept, whereas the second represents an estimation of the student's proneness to make mistakes while using this concept. The values of the estimations are within the range [0-1]. When estimating the degree of knowledge of a particular concept, 0 depicts the system's belief that the student does not know the concept at all, while 1 represents the system's belief that the student knows this concept very well. Furthermore, when referring to error proneness concerning a concept, 0 represents the system's belief that the student never makes mistakes when using a concept, and 1 represents the system's belief that the student always makes mistakes related to the concept. Therefore, the second vector is defined as shown in the example of Section 3.

Web-PVT produces the second vector for a new student taking into account the characteristics of the first student model vector of the new student as well as of the other students who belong to the same stereotype concerning their knowledge level. The distance between students is calculated as shown in Equation 3, and the attributes that are used to define this distance are the degree of carefulness of students, their mother tongue and the other languages that they already know.

6. Evaluation of the Initial Student Models in the Web-based ICALL

Evaluation of student models is very important because it may reveal whether a student modeler is effective or not. Despite the importance of evaluations, there is a shortage of them in user modeling systems. Chin (2001) in a review of user modeling articles points out that there are insufficient empirical evaluations. In order to assess the effectiveness of the ISM framework concerning the initialization of student models in Web-PVT, we conducted an evaluation study. The aim of the student modeler is to produce more individualized initial student models, as the system learns the models of other similar students. Therefore, we investigated the accuracy of the predicted initial student models at different points of the system's usage. The method that we used was by comparison of student models with human experts' beliefs about the students in question, in a similar way as in (Virvou & DuBoulay, 1999).

In particular, three teachers and their classes were asked to participate in the experiment. The teachers taught their students for at least a whole school year. Therefore, they knew their students' abilities from the lessons and thus they had formed some beliefs about them. At the evaluation they had to compare their beliefs about their students with the assumptions generated by the system about them. In particular, the teachers were presented with the initial models (feature vectors) of their students as they were produced by Web-PVT and they were asked to provide an agreement rate for each initial student model. Each teacher evaluated only the models of students who belonged to their classes. This was done because teachers who had never taught a particular student were not expected to have formed beliefs about this student's knowledge and misconceptions. The experiment was conducted in two phases. In the first phase, teachers assessed the initial models of students that were generated based on their classification into stereotypes. Thus, teachers were asked to evaluate the accuracy of the default assumptions of stereotypes for the students categorized in each one of them. The second phase, took place after 15 students of each stereotype had already interacted with the system sufficiently for it to have made observations about their behavior, which was recorded in their individual student models. The number of interactions was considered sufficient when two conditions were satisfied: (a) the system had the opportunity to make observations that resulted in altering the default value of at least one characteristic of the student model, and (b) the particular student still belonged to the same stereotype; this means that the student had not interacted with the system long enough to have acquired knowledge that would place her/him to a more advanced stereotype. Both the above conditions were usually satisfied when students had interacted with Web-PVT for not more than three hours. In this way, the system already knew about the 15 students of each stereotype and it could use this knowledge to derive inferences about new students. Then, five new students of each stereotype were asked to register into the system and their initial student models, that were generated using the ISM framework, were evaluated by their teachers. The experiment was completed by the comparison of the results of the first phase with those of the second phase.

More specifically, three teachers of English and their students (117 students) participated in the experiment. At first, each teacher categorized her/his students in one of the four stereotype categories that were supported by Web-PVT. Following this process, among the 117 students, 20 belonged to the novice stereotype, 38 to the beginner stereotype, 36 to the intermediate stereotype and 23 to the advanced stereotype. This information was then used for the selection of the students that would participate in each phase of the evaluation experiment; for example, for randomly selecting 15 students among the students of each stereotype category that would use the system for some time. Then, teachers were asked to evaluate the accuracy of the initial models of their students as they were produced by the student modeler of Web-PVT. More specifically, teachers were asked to evaluate five randomly chosen initial student models from each one of the four supported stereotypes at two phases. At the first phase, the teachers evaluated five initial models of students belonging to each one of the supported stereotypes (a sum of 20 initial student models), before any student had been registered to the system. At this phase, the student modeler of Web-PVT did not have any knowledge about other students who were similar to the new students. Therefore, the initial student models evaluated in this phase were produced by using only the default assumptions of the supported stereotypes. For this reason, the evaluation at this stage could be considered as an evaluation of the stereotypes that were hand-coded to Web-PVT.

At the second phase of the study, teachers evaluated the initial models of 5 randomly chosen students of each stereotype, after Web-PVT had constructed the models of 15 students of each stereotype, based on direct observations of the students' interaction behavior. For this reason, 15 students of each stereotype category (a sum of 60 students) were registered to the system and they were asked to interact with it for about two hours in order to study theory and solve exercises. For reasons of fair comparison of the results for each one of the stereotype categories, some students were omitted from the study. In particular, we wanted to evaluate the initial student models of students of each stereotype based on the models of the same number of similar students. Thus, for the production of the initial models in the second phase, we used only 15 students of each stereotype for the system to learn about their behavior and 5 students for the system to produce initial models, since the novice stereotype consisted of 20 students only.

In both phases of the evaluation, for each one of the initial student models produced by the student modeler of Web-PVT, the teacher who was responsible for the particular student was asked to provide a percentage of her/his agreement (0% indicating that the teacher totally disagrees and 100% indicating that the teacher totally agrees) with the estimations of the system about the knowledge level and the error proneness of the student for each domain concept. Then, the teacher's overall agreement with the system's initial model was calculated as the mean value of all agreement percentages. The experimental hypothesis was that an initial student model that was built after the system had constructed models of other students of the same knowledge level stereotype would be superior to the initial student model constructed by using the default assumptions of the stereotype that becomes active for a specific student. In order to evaluate the hypothesis, we used a one-tailed paired t-test with the alpha level set to 0.05. Table II presents the results of the evaluation study.

A first conclusion that could be drawn based on the results of the evaluation study is that the inferences of the stereotypes that were hand-crafted to Web-PVT

Table II. Statistical analysis of the hypothesis (one-tailed paired t-test)

Stereotype	Mean value of teacher's agreement after 0 students of the same stereotype have used Web-PVT (First phase)	Mean value of teacher's agreement after 15 students of the same stereotype have used Web-PVT (Second phase)	p value	t value (df)
Novice	78.4%	83.0%	0.0383	−2.3722 (4)
Beginner	84.6%	90.2%	0.0124	−3.5 (4)
Intermediate	89.2%	91.2%	0.0108	−3.6515 (4)
Advanced	92.0%	92.4%	0.3946	−0.2857 (4)

seemed to have achieved a high degree of acceptance from the teachers. However, the student modeler in all the cases performed better at initializing the model of a new student taking into account other students of the same knowledge level stereotype than when it used the stereotype default assumptions. Therefore, ISM managed to produce more accurate initial student models for all stereotype categories of students. In most of the cases, the initial student models that were generated when similar students had been found achieved statistically significant higher acceptance from teachers than those produced when no similar students were found in the student model knowledge base. For example, the teachers' agreement rate with the initial models of students that belonged to the intermediate stereotype was increased from 89.2% at the first phase of the evaluation to 91.2% at the second phase. This increase was statistically significant ($p = 0.0108$). The only case where there was no statistically significant increase in the accuracy of the initial models of students concerned the advanced stereotype ($p = 0.3946$). This result could be explained based on the fact that advanced students do not make many mistakes and therefore, the errors due to language transfer are significantly reduced. In view of the above, we can conclude that the mother tongue of advanced students and the other languages they may know does not play a significant role in explaining their knowledge level and error proneness in the domain of the passive voice of the English language. Moreover, the acceptance of the default assumptions of the stereotype by teachers was already very high and thus there was not much scope for improvement. However, in general terms, ISM managed to produce more accurate initial student models than the stereotype categories alone.

In fact, the results gained by the k-NN calculations increase the accuracies reported by 5%, 6%, 2% and 0.4%. These differences are statistically significant but they may not look impressively high. However, they are actually very meaningful. This is so because in real conditions where Web-PVT will be used by many remote users of many different backgrounds for a long period of time, the system is expected to perform even better than in the evaluation. Indeed, the system will gain more experience from more users and will be able to classify them more accurately if they have different characteristics. For example, in a situation where users would have many different mother tongues and they would know many different foreign languages, the system would have the opportunity to compute stronger and weaker similarities that would provide more accurate classifications of new students. The current study could only be applied to a smaller and less diverse student population. Even so, the results are very promising.

Furthermore, the initialization of student models using ISM reduced the amount of time that students should spend in the preliminary test as compared to a thorough test that would contain questions related to all the concepts of the domain knowledge of Web-PVT. Indeed, the preliminary test used by Web-PVT to assess the knowledge level of students contains ten questions that are usually answered in fifteen minutes. A thorough preliminary test, on the other hand, would require the student to answer to at least forty-two questions (one question for each concept of the domain) that would take her/him about forty-five minutes.

7. Conclusions and Future Work

In this paper we have described a framework that addresses the problem of the initialization of student models in Web-based educational applications. Our approach to student model initialization exploits the fact that Web-based systems have a large number of users and we use a machine learning reasoning mechanism that is based on recognized similarities between users. The initialization of the student model is performed dynamically for each student taking into account information that originates from other students' performance while using the system. In this way the initialization procedure is automatically updated each time a student interacts with the system.

In particular, we have created a framework called ISM, that makes use of a novel combination of stereotypes and the distance weighted k-NN algorithm in order to set initial values to the model of a new student. Stereotypes are used to make initial hypotheses about the knowledge level of the student, whereas the distance weighted k-NN algorithm is utilized to refine the estimations of the student's knowledge level of each concept and her/his proneness to make mistakes concerning this concept, based on the student's similarity with other students of the same stereotype category. The similarity between students is estimated based on the student characteristics that may play a role in the student's performance while s/he learns the domain being taught by the application.

The initialization method of the ISM framework could be adapted by different ITSs, by defining the student characteristics that will be used to measure the distance between students, and that should be acquired in the initial phase of the system's usage. The exact student characteristics that should be taken into account in the ISM framework should be specified by following the instructions of domain experts or according to the results of empirical studies that would involve domain experts or based on domain expertise that is based on the didactics of a particular domain.

The ISM framework has been evaluated on its ability to generate beliefs about new students as compared with human tutors. In particular, the potential success of the student models has been evaluated in an empirical study that was conducted using Web-PVT, which was based on the ISM framework. The results of this evaluation showed that with the use of the ISM framework more detailed student models could be built more quickly as opposed to the non use of this framework. This is considered a very good result, given the fact that the evaluation was conducted among a relatively small number of students as compared to the potential number of Web users that Web-PVT could accommodate. An increased number of users of diverse backgrounds is expected to yield even better results since the system will have more students to learn from and will be able to produce more accurate classifications of the newcomers.

The ISM framework has been successful in generating initial student models in Web-PVT. However, it must be noted that an exhaustive and very detailed evaluation of the ISM framework would involve more experiments that would need a long time to complete. In particular, first we should evaluate the framework with respect to its generality and domain independence. Indeed, we have already implemented the methodology of ISM in a Web-based ITS for the domain of algebra (Tsiriga and Virvou, 2002b). The selection of the domain of algebra was based on the fact that it is very different from the domain of language. Indeed, the two domains have totally different factors that are considered important for the learning process. Hence the successful application of our framework on such a different domain would be considered a good test for its generality. However, we would need to conduct a full evaluation of this system in order to reveal whether ISM is successful in this domain too. Furthermore, we would also need to evaluate Web-PVT for a long period of time, using many remote students over the Web to confirm the fact that the experience gained by the system will improve the results significantly. Finally, we also plan on evaluating the impact of the improved student modeling approach in providing personalized instruction more quickly. In particular, we should conduct evaluation studies to examine whether the initial student models produced by the use of ISM lead to more individualized tutoring in the students' first interactions with the system as compared to the student models produced by using stereotypes alone.

Acknowledgements

The authors would like to thank the anonymous reviewers and the editor for their careful reading of the paper and for their detailed and helpful comments.

References

Aha, D., Kibler, D. and Albert, M.: 1991, Instance based learning algorithms. *Machine Learning* **6**, 37–66.

Aïmeur, E., Blanchard, E., Brassard, G. and Gamps, S.: 2001, QUANTI: a multidisciplinary knowledge-based system for quantum information processing. In: *Proceedings of the International Conference on Computer Aided Learning in Engineering Education* (CALIE'01), pp. 51–57.

Aïmeur, E., Brassard, G., Dufort, H. and Gamps, S.: 2002, CLARISSE: a machine learning tool to initialize student models. In: S. A. Cerri, G. Gouardéres and F. Paraguaçu (eds.): *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Vol. 2363. Springer-Verlag, Berlin, Heidelberg, pp. 718–728.

Albrecht, F., Koch, N. and Tiller, T.: 2000, SmexWeb: an adaptive web-based hypermedia teaching system. *Journal of Interactive Learning Research, Special Issue on Intelligent Systems/Tools in Training and Lifelong Learning* **11**(3/4), 367–388.

Alpert, S. R., Singley, M. K. and Fairweather, P. G.: 1999, Deploying intelligent tutors on the web: an architecture and an example. *Journal of Artificial Intelligence in Education* **10**, 183–197.

Baffes, P. and Mooney, R.: 1996, Refinement-based student modeling and automated bug library construction. *Journal of Artificial Intelligence in Education* **7**(1), 75–116.

Beck, J. and Woolf, B.: 2000, High-level student modeling with machine learning. In: G. Gauthier, C. Frasson and K. VanLehn (eds.): *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Vol. 1839. Springer-Verlag, Berlin, Heidelberg, pp. 584–593.

Beeson, M.: 1989, The user model in MATHPERT: an expert system for learning mathematics. In: D. Bierman, J. Breuker and J. Sandberg (eds.): *Proceedings of the Fourth International Conference on Artificial Intelligence and Education*. IOS, Amsterdam, pp. 9–14.

Bontcheva, K.: 2002, Adaptivity, adaptability, and reading behaviour: some results from the evaluation of a dynamic hypertext system, In: P. De Bra, P. Brusilovsky and R. Conejo (eds.): *Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science*, Vol. 2347. Springer-Verlag, Berlin, Heidelberg, pp. 69–78.

Brusilovsky, P.: 1996, Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction* **6**(2/3), 87–129

Brusilovsky, P. and Pesin, L.: 1998, Adaptive navigation support in educational hypermedia: an evaluation of the ISIS tutor. *Journal of Computing and Information Technology* **6**(1), 27–38.

Burton, R.: 1982, Diagnosing bugs in a simple procedural skill. In: D. Sleeman and L. Brown (eds.): *Intelligent Tutoring Systems*. Academic Press, London, pp. 157–183.

Chin, D.: 2001, Empirical evaluation of user models and user adapted systems. *User Modeling and User-Adapted Interaction* **11**(1/2), 181–194.

Chiu, B. C. and Webb, G. I.: 1998, Using decision trees for agent modeling: improving prediction performance. *User Modeling and User-Adapted Interaction* **8**(1/2), 131–152.

Cover, T. and Hart, P.: 1967, Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* **13**(1), 21–27.

Dasarathy, B.: 1991, *Nearest Neighbor* (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos CA.

da Silva, P., Van Durm, R., Hendrikx, K., Duval, E. and Olivie, H.: 1997, A simple model for adaptive courseware navigation. In: S. Lobodzinski and I. Tomek (eds.): *Proceedings of WebNet '97, World Conference of the WWW, Internet and Intranet*. AACE, Charlotsville, pp. 959–960.

da Silva, P., Van Durm, R., Duval, E. and Olivie, H.: 1998, Concepts and documents for adaptive educational hypermedia: a model and a prototype. In: P. Brusilovsky and P. De Bra (eds.): *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia*, pp. 35–43.

De Bra, P.: 2000, Pros and cons of adaptive hypermedia in web-based education. *Journal of CyberPsychology and Behavior* **3**(1), 71–77.

Dudani, S.: 1976, The distance-weighted *k*-nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics* **6**(4), 325–327.

Emde, W. and Wettshereck, D.: 1996, Relational instance-based learning. In: L. Saitta (ed.): *Proceedings of the Thirteenth International Conference on Machine Learning*. Morgan Kaufmann, San Mateo CA, pp. 122–130.

Garofalakis, J., Sakkopoulos, E., Sirmakessis, S. and Tsakalidis, A.: 2002, Integrating adaptive techniques into virtual university learning environment. In: V. Petrushin, P. Kommers, Kinshuk and I. Galeev (eds.): *Proceedings of the 2002 IEEE International Conference on Advanced Learning Technologies: Media and the Culture of Learning*. IEEE Computer Society, Palmerston North New Zealand, pp. 28–33.

Gürer, D., des Jardins, M. and Schlager, M.: 1995, Representing a student's learning states and transitions. In: M. T. Cox and M. Freed (eds.): *Proceedings of the 1995 Spring Symposium on Representing Mental States and Mechanisms*. AAAI Press, Menlo Park CA, pp. 51–59.

Guzman, E. and Conejo, R.: 2002, Simultaneous evaluation of multiple topics in SIETTE. In: S. A. Cerri, G. Gouardéres and F. Paraguaçu (eds.): *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Vol. 2363. Springer-Verlag, Berlin, Heidelberg, pp. 739–748.

Heift, T. and Nicholson, D.: 2000, Theoretical and practical considerations for web-based intelligent language tutoring systems. In: G. Gauthier, C. Frasson and K. VanLehn (eds.): *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Vol. 1839. Springer-Verlag, Berlin, Heidelberg, pp. 354–363.

Heift, T. and Nicholson, D.: 2001, Web delivery of adaptive and interactive language tutoring. *International Journal of Artificial Intelligence in Education* **12**, 310–324.

Henze, N. and Nejdl, W.: 2001, Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education* **12**, 325–350.

Hoppe, U.: 1994, Deductive error diagnosis and inductive error generalization for intelligent tutoring systems. *Journal of Artificial Intelligence in Education* **5**(1), 27–49.

Hothi, J. and Hall, W.: 1998, An evaluation of adaptive hypermedia techniques using static user modeling. In: P. Brusilovsky and P. De Bra (eds.): *Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia*, pp. 45–55.

Kay, J.: 2000, Stereotypes, student models and scrutability. In: G. Gauthier, C. Frasson and K. VanLehn (eds.): *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Vol. 1839. Springer-Verlag, Berlin, Heidelberg, pp. 19–30.

Kobsa, A., Koenemann, J. and Pohl, W.: 2001, Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review* **16**(2), 111–155.

Kurhila, J., Miettinen, M., Niemivirta, M., Nokelainen, P., Silander, T. and Tirri, H.: 2001, Bayesian modeling in an adaptive on-line questionnaire for education and educational research. In: H. Ruokamo, O. Nykanen, S. Pohjolainen and P. Hietala (eds.): *Proceedings of the Tenth International PEG Conference*, pp. 194–201.

MacLeod, J., Luk, A. and Titterington, D.: 1987, A re-examination of the distance-weighted *k*-nearest-neighbor classification rule. *IEEE Transactions on Systems, Man and Cybernetics* **17**(4), 689–696.

Mitchell, T.: 1997, *Machine Learning*. McGraw-Hill, New York.

Moriarty, C., Kushmerick, N. and Smyth, B.: 2001, Personalized intelligent tutoring for digital libraries. In: *Proceedings of the Second DELOS Network of Excellence Workshop on Personalization and Recommender Systems in Digital Libraries*.

Murphy, M. and McTear, M.: 1997, Learner modelling for intelligent CALL. In: A. Jameson, C. Paris and C. Tasso (eds.): *Proceedings of the Sixth International Conference on User Modeling*. Springer, Vienna New York, pp. 301–312.

Nwana, H.: 1991, User modelling and user adapted interaction in an intelligent tutoring system. *User Modeling and User-Adapted Interaction* **1**(1), 1–32.

Ogata, H., Liu, Y., Ochi, Y. and Yano, Y.: 2001, Neclle: network-based communicative language learning environment focusing on communicative gaps. *Computers and Education* **37**, 225–240.

Okazaki, Y., Watanabe, K. and Kondo, H.: 1996, An implementation of an intelligent tutoring system (ITS) on the World-Wide Web (WWW). *Educational Technology Research* **19**(1), 35–44.

Paliouras, G., Karkaletsis, V., Papatheodorou, C. and Spyropoulos, C. D.: 1999, Exploiting learning techniques for the acquisition of user stereotypes and communities. In: J. Kay (ed.): *Proceedings of the Seventh International Conference on User Modeling, CISM Courses and Lectures*, Vol. 407. Springer, Vienna New York, pp. 169–178.

Rich, E.: 1979, User modelling via stereotypes. *Cognitive Science* **3**(4), 329–354.

Rich, E.: 1983, Users are individuals: individualizing user models. *International Journal of Man-Machine Studies* **18**, 199–214.

Schwab, I. and Kobsa, A.: 2002, Adaptivity through unobstructive learning. *Künstliche Intelligenz* **3**, 5–9.

Sison, R., Numao, M. and Shimura, M.: 1998, Discovering error classes from discrepancies in novice behaviors via multistrategy conceptual clustering. *User Modeling and User Adapted Interaction* **8**(1/2), 103–129.

Sison, R., Numao, M. and Shimura, M.: 2000, Multistrategy discovery and detection of novice programmer errors. *Machine Learning* **38**, 157–180.

Sison, R. and Shimura, M.: 1998, Student modeling and machine learning. *International Journal of Artificial Intelligence in Education* **9**, 128–158.

Sleeman, D.: 1987, PIXIE: a shell for developing intelligent tutoring systems. In: R. Lawer and M. Yazdani (eds.): *Artificial Intelligence in Education*. Ablex, New Jersey.

Tchètagni, J. and Nkambou, R.: 2002, Hierarchical representation and evaluation of the student in an intelligent tutoring system. In: S. A. Cerri, G. Gouardéres and F. Paraguaçu (eds.): *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Vol. 2363. Springer-Verlag, Berlin Heidelberg, pp. 708–717.

Tsiriga, V. and Virvou, M.: 2002a, Dynamically initializing the student model in a web-based language tutor. In: *Proceedings of the 2002 First International IEEE Symposium 'Intelligent Systems'*, Vol. I. IEEE Computer Society Press, pp. 138–143.

Tsiriga, V. and Virvou, M.: 2002b, Initializing the student model using stereotypes and machine learning. In: A. El Kamel, K. Mellouli and P. Borne (eds.): *Proceedings of the 2002 IEEE International Conference on System, Man and Cybernetics*.

Vassileva, J.: 1997, Dynamic course generation on the WWW. In: B. du Boulay and R. Mizoguchi (eds.): *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. IOS, Amsterdam, pp. 498–505.

Virvou, M. and DuBoulay, B.: 1999, Human plausible reasoning for intelligent help. *User Modeling and User-Adapted Interaction* **9**(4), 321–375.

Virvou, M. and Moundridou, M.: 2001, Student and instructor models: two kinds of user model and their interaction in an ITS authoring tool. In: M. Bauer, P. Gmytrasiewicz and J. Vassileva (eds.): *Proceedings of the Eighth International Conference on User Modeling, Lecture Notes in Artificial Intelligence*, Vol. 2109. Springer-Verlag, Berlin Heidelberg, pp. 158–167.

Virvou, M. and Tsiriga, V.: 2001, Web passive voice tutor: an intelligent computer assisted language learning system over the WWW. In: T. Okamoto, R. Hartley, Kinshuk and J. Klus (eds.): *Proceedings of the IEEE International Conference on Advanced Learning Technologies: Issues, Achievements and Challenges*. IEEE Computer Society Press, Los Alamitos CA, pp. 131–134.

Weber, G. and Specht, M.: 1997, User modeling and adaptive navigation support in WWW-based tutoring systems. In: A. Jameson, C. Paris and C. Tasso (eds.): *Proceedings of the Sixth International Conference on User Modeling*. Springer, Vienna New York, pp. 289–300.

Wilson, R. and Martinez, T.: 1997, Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research* **6**, 1–34.

Authors' vitae

Victoria Tsiriga received her Ph.D. in the area of student modeling in Web-based education from the Department of Informatics, University of Piraeus. She graduated from the same Department in 1997. Her current research interests lie in the areas of user modeling, web-based intelligent tutoring systems and human-computer interaction.

Maria Virvou is an assistant professor in the Department of Informatics at the University of Piraeus, Greece. She received her Ph.D. in Artificial Intelligence and Computer Science at the University of Sussex and an M.Sc. degree in Computer Science from University College London. Her first degree in Mathematics was obtained from the University of Athens, Greece. Her current research interests include user modeling, adaptive and intelligent user interfaces, knowledge-based software engineering, object-oriented software engineering, artificial intelligence in education and e-learning.