Data-Driven Student Knowledge Assessment through Ill-Defined Procedural Tasks

Jaime Gálvez, Eduardo Guzmán, and Ricardo Conejo

Dpto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga,
Bulevar Louis Pasteur, 35, Campus de Teatinos,
29071 Málaga, Spain
{jgalvez,guzman,conejo}@lcc.uma.es

Abstract. The Item Response Theory (IRT) is a statistical mechanism successfully used since the beginning of the 20th century to infer student knowledge through tests. Nevertheless, existing well-founded techniques to assess procedural tasks are generally complex and applied to well-defined tasks. In this paper, we describe how, using a set of techniques we have developed based on IRT, it is possible to infer declarative student knowledge through procedural tasks. We describe how these techniques have been used with undergraduate students, in the object oriented programming domain, through ill-defined procedural exercises.

Keywords: Student modeling, student assessment.

1 Introduction

Since the birth of the first teaching systems, Artificial Intelligence (AI) techniques have been used in order to try to provide those systems with the required capabilities to emulate human tutors [1]. Knowledge representation, student modeling and cognitive diagnosis (or student knowledge inference) are only some of the capabilities involved in the development of education systems. AI is used in this context to offer students a personalized learning process according to their needs, adapting to them in order to influence them in as positive a way as possible.

From the student modeling perspective, several techniques for representing (and diagnosing) student characteristics, such as his/her knowledge level, his/her conceptual errors, etc., can be found in the literature (cf. [2]). Nowadays, there exist well-founded assessment mechanisms, based on statistical theories such as *the Item Response Theory* (IRT) [3]. This theory is used successfully to infer the student knowledge using tests. Nevertheless, the use of existing well-founded techniques to assess the knowledge using procedural activities such as problems, are generally complex. In this context, proposals such as *Model Tracing* [4] or others based on Bayesian networks [5], involve a great effort to model the procedural activities, since the set of steps the student could carry out should be defined a priori [6]; and therefore, the modeling of those domains becomes an arduous task.

In preliminary studies [7] we used well-founded techniques to evaluate how we can perform an assessment of the student knowledge in a domain with well-defined tasks:

P. Meseguer, L. Mandow, and R.M. Gasca (Eds.): CAEPIA 2009, LNAI 5988, pp. 233–241, 2010.

this assessment is the result of the application of the Simplex optimization algorithm. In the paper cited, techniques based on the IRT were described, and its suitability was evaluated. In this paper we focus on a domain where the solution space makes tasks be ill-defined. The hypothesis from which we start is that, using the proposed techniques, the student's declarative knowledge inferences are equivalent to those that would be provided using an IRT-based test. The strength of our proposal is that much less problems are required to achieve an accurate assessment diagnosis.

The article is structured as follows: In the next section, the domain modeling technique that we use in our approach, called Constraint-Based Modeling (CBM), is described. Next, we will explain the fundamentals of the IRT. In section 4, the key ideas of this work are presented. Section 5 focuses on explaining the more relevant features of the educational tool we have developed. Subsequently, in section 6, the experiment performed with the educational tool to verify our hypothesis is described, together with the results obtained. Finally, conclusions and future work are presented.

2 Constraint-Based Modeling

The use of CBM in the learning environment contributes to improve the student learning process, making students learn from their own mistakes when solving a problem from a particular domain. The CBM is based on Ohlsson's theory [8] of learning from performance errors. According to it, learning is a two-step process: In the first, an error is detected while an activity is being performed; and afterwards, it is corrected. Errors may take place when the students try to solve a problem but either they do not have the required declarative knowledge or they are unable to apply the procedure appropriately.

To detect errors, CBM-based tutor systems generate a representation of the solution the student is building, which is updated according to the actions performed by the student in the system interface. In CBM, the domain is represented by a set of principles, which will be compared with the student solution representation and in this way it will be inferred which domain principles are being violated by that solution. The principles that form the domain are considered to be the basic unit of knowledge in the CBM and they are represented by constraints about the state that must be satisfied by all possible correct solutions of all problems. In other words, a correct solution to a problem will never generate a representation that violates some of the constraints of the domain.

According to CBM, each constraint is defined by an ordered pair of conditions: C_r, C_s; where C_r is the relevant condition, which is employed to determine the sort of problems and states for which the constraint is relevant, thus, where it could be applied. C_s is the satisfaction condition and contains the error condition associated to a certain principle that a solution for a given problem could infringe. When the C_r of a constraint is true, for a certain state of a problem solution, it is said that the constraint is significant, from the pedagogical point of view, and therefore, C_s should also be true. Otherwise, the constraint has been violated, which implies that one or more errors have occurred. After detecting those errors, the student model is updated and the system should provide the student with some feedback and apply some corrective action that helps the student to correct his/her conceptual mistake.

The results obtained by several tutors based on the CBM prove the effectiveness of this approach in learning tasks [9] and its suitability as compared with other similar proposals. Nevertheless, Ohlsson and Mitrovic [10] have remarked that, to allow the systems that use the CBM be able to help when taking pedagogical decisions, it is necessary to have a long term student model. In that sense, most of the existing proposals based on CBM (with the exception of some that include Bayesian networks [11]), infer the student knowledge as a proportion of the constraints the student knows. However, this heuristic does not have characteristics that are mandatory in a knowledge diagnosis system (and in general for every system of this type), such as the invariance. For this reason, estimations based on heuristics are strongly conditioned by the particular problems the student has made.

3 Item Response Theory

The IRT, developed by Thurstone [3], is the most popular discipline, based on statistical techniques, for quantitatively measuring certain traits such as the intelligence, skills and/or, knowledge level in a given concept, personality, etc. This theory is based on two principles [12]: According to the first, the knowledge that a student has in a test question (or *item*) can be quantified through a factor called *knowledge level*. The second principle establishes that the relationship between the probability of answering an item correctly and the student knowledge level can be described by means of a monotonically increasing function named *Item Characteristic Curve* (ICC). The higher the student knowledge level, the higher the probability of answering correctly. This function is the central element of the IRT. One of the most frequently used functions (and perhaps, the most popular) to model the ICC is the three-parameter logistic function (3PL):

$$P(u = 1|\theta) = c_i + (1 - c_i)\frac{1}{1 + e^{-1.7a_i(\theta - b_i)}} \quad (1)$$

where $P(u_i = 1| \theta)$ is the probability of answering correctly the item i given the student knowledge level θ, which is normally measured using a continuous scale between $[-3.0, .., 3.0]$. The three parameters that characterize this curve depend on the item and they are:

- The *discrimination* (a_i), which is a value proportional to the slope of the curve and the higher it is, the more the item discerns between the inferior and superior knowledge levels.
- The *difficulty* (b_i), which corresponds to the value of the knowledge level for which the probability of answering correctly is the same as that of answering incorrectly (without taking into account randomly selected responses).
- The *guessing* (c_i), which measures precisely the probability that a student will answer correctly even though he/she may not possess the knowledge required to do so, modeling thus, those situations where the student answers randomly.

The popularity of the IRT is a direct consequence of the consistence of its results. In other proposals such as the *Classic Test Theory*, the results of estimating the student

knowledge depend on the sample of students used to perform the test and, thus, the results in the test are not comparable to those obtained in other different tests. The results obtained by applying the IRT however posses several properties such as the invariance. In other words, the knowledge level inferred using this approach does not depend on the test. Therefore, if two tests which assess the same concept, are administered to the same student, the results obtained would be very similar.

In order to apply the IRT it is necessary to have available the ICC values corresponding to each item in the domain. To achieve this, a data driven procedure called *calibration* is required. Calibration is a statistical process for which data must be available on a student population sample previously evaluated / tested. Through this procedure, the parameters that characterize each ICC are inferred. The input to this procedure is based on the results from those students who did tests using the questions whose curves we wish to infer.

4 Assessment Combining the IRT and CBM through Procedural Tasks

Through the IRT, student knowledge inferences can be made with desirable characteristics such as the invariance. Nevertheless, in theory, the IRT is difficult to apply when the goal is to assess activities of procedural type. Indeed, to carry out an assessment similar to that made by a teacher for a problem solved by a student, but using the IRT, it would be necessary to build a very large number of items that were focused on all the issues that could be evaluated with only one problem.

Our proposal aims to solve the inconveniences that present both the IRT and the CBM, by means of a set of assessment techniques combining the two approaches. The goal is to improve the heuristics that are normally used in the CBM for updating the long term student model. This improvement consists of introducing inference techniques of the student knowledge inspired by IRT fundamentals. Therefore, the evidence that the student provides about his/her knowledge will be the actions that he/she performs while resolving a problem. Those actions will be translated into violations (or satisfactions) of constraints from a set of constraints used to represent the domain matter.

In the IRT the elements used to determine the student knowledge are the items, however in our proposal the constraints are used. Thus, every constraint will have associated a characteristic curve that we have named *Constraint Characteristic Curve* (CCC). This curve has the opposite shape to an ICC since, while this last represents the probability of answering correctly (knowledge), the CCC represents the probability of violating a constraint in a given problem (detection of incorrect knowledge). When a constraint is violated, this implies a lack of knowledge and, therefore, the necessary curve to represent it must monotonically decrease. The higher the student knowledge level, the lower the probability of that constraint being violated. In the IRT, it would be equivalent to a wrong response to an item.

The student knowledge distribution $P(\theta \mid \Phi, \tau)$ would be calculated as the product of the CCCs of those constraints that have been violated, joined with the opposite curves for those that, being relevant for the problem, have not been violated. This form of calculating the student knowledge is based on the inference mechanisms used in the IRT:

$$P(u = 1|\theta) = c_i + (1 - c_i)\frac{1}{1 + e^{-1.7a_i(\theta - b_i)}} \quad (2)$$

where $\Phi = p_1, p_2, ..., p_m$ represents the set of problems solved by the student and $\tau = c_1, c_2, ..., c_n$ the collection of constraints pertaining to the domain. $P(c_j \mid \theta)$ is the characteristic curve of the constraint c_j; r_{ij} is a binary variable that indicates whether or not the constraint is relevant to the problem p_i; and f_{ij} is another binary variable that indicates whether the student action in the problem p_i has produced the violation of the constraint c_j.

In our previous work [13] we used discrete curves whose values are the pairs knowledge level / probability, simplifying noticeably the necessity of data required by the IRT to infer the characteristic curves. In this proposal, the CCCs are also discrete and every value indicates the probability of a student with certain knowledge level violating a constraint.

The result of applying equation 2 is a distribution of probabilities where, for each level of knowledge, the probability of the student knowledge level corresponding to that level is expressed. In order to calculate the denominated knowledge level two strategies can be applied: taking the distribution's expected value (or mean) (*Expectation a Posteriori*), or also, choosing the mode (*Maximum a Posteriori*).

5 OOPS

To put into practice the model proposed in the previous section, we have used a new version of OOPS (Object Oriented Programming System) [14]. This new version incorporates fundamentals elements of CBM. OOPS focuses on the Object-Oriented programming domain and allows emulating the behavior of a human tutor during the student learning process, detecting the gaps in their knowledge and acting to rectify the situation. This tool permits the students to construct object-oriented programs in the pseudo-language used in the School of Telecommunication Engineering at the University of Málaga (Spain) as a result of doing exercises. These exercises are based on defining and implementing classes (attributes and methods), according to a stem provided by the system. The construction is done visually using the drag and drop technique. The student has to select from a toolbar the elements needed to construct an object-oriented program, and drop them into the workspace.

Once the student decides to correct his/her solution, the system will initialize its inference engine, which uses the domain model constraints for detecting the student's errors. With these errors, the student model is updated and, unless the system is working in evaluation mode, the students will be shown feedback to help them to correct their conceptual error.

The current architecture of OOPS is based on a generic framework [15] defined for the construction of Web-based Intelligent Learning Environments, based on problem solving tasks. For this reason, OOPS is structured in the following modules:

1. The most external component, the *interface* (see Figure 1), through which teachers and students can insert or solve problems, respectively. Teachers can configure the system to show information such as the hint/feedback messages.

2. The main part of OOPS is the *pedagogical module*, which has the components required for estimating the student knowledge, such as the estimation algorithms, a JESS inference engine [16] for identifying the violated constraints, or those elements needed to control the pedagogical actions used to instruct the student in the most suitable way, such as the adaptive problem selection module.
3. The *domain model*, formed by the set of constraints and problems.
4. The *student model*, which contains the violated and the satisfied constraints, the knowledge estimation distributions and the logs with the actions performed by the student in each problem.

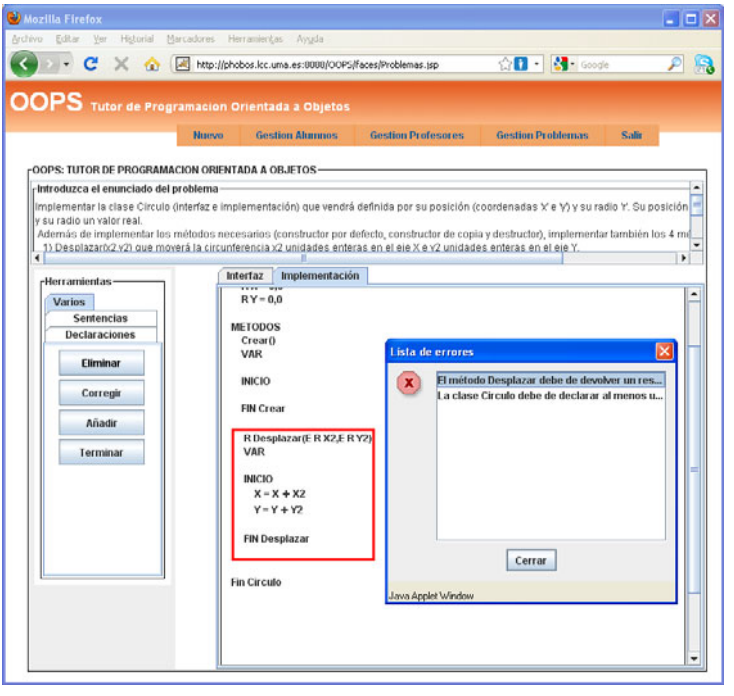

Fig. 1. OOPS interface

6 Experiments

To check the validity of our diagnostic technique through ill-defined procedural tasks, we carried out an experiment with undergraduate students. Our goal was to allow the students to put into practice their knowledge of object-oriented programming in a session which took place in a controlled environment (in a teaching laboratory). The session was structured as follows: First, a test was administered using the Siette web-based system [17], which provides IRT-based assessments. Next, students used OOPS to solve two problems.

We stated the following hypothesis: if we infer the student knowledge level through IRT tests (in this experiment through Siette) and with our model (implemented in OOPS), the results obtained should be similar.

6.1 Experiment Design

The experiment was design in such a way that in both parts (the test and the problems) the same kind of knowledge was measured. Accordingly, the test questions were elaborated carefully to assess the same concepts used in the practical problems proposed in OOPS. For this purpose, in OOPS two problems were selected involving basic concepts of object-oriented programming. That is, the most relevant domain constraints were checked in those problems. We used a subset of 15 significant constraints, according to some domain experts. The set of questions were developed taking into account the subset of constraints. Two test questions were created for each constraint. As a result, these questions assessed the same concept the constraint gauged and this led to a total of 30 test questions.

The experiment was carried out in May of 2009 with undergraduate students studying Technical Engineering in Telecommunications from the University of Málaga (Spain). These students had previously received face-to-face lessons on object-oriented programming. A total of 20 students participated in the experiment. After being administered the test (where the solutions were never shown), the students started using OOPS. Initially, all of them took a problem (whose results were not considered in the posterior analysis) simply for training on the system. Next, they took two programming problems in OOPS.

6.2 Data Analysis

Once the data from both systems were obtained, we processed this information using a tool called MULTILOG [18], which is one of the most popular for IRT-based analysis. First, we calibrated the ICCs according to the 3PL IRT-based model. To this end, we used the test results represented in a matrix of boolean values. This matrix, needed by MULTILOG, contained, for each student and item (question), a value indicating whether or not the concept evaluated was known by the student. Subsequentyly, the obtained calibration was used in conjunction with the test results to infer the student knowledge estimation.

Analogously, data obtained through OOPS were used to calibrate the CCCs. In this case, the matrix used by MULTILOG represented whether or not the constraint was violated (that is, whether or not the concept is known) during the problem resolution. The curves computed after this step, were used again with the student data collected by OOPS, to generate the student knowledge estimations.

Our main aim was to obtain homogeneous results in relation to the knowledge source assessed and the nature of the data used to estimate this knowledge. Regarding the knowledge source, we used the previously mentioned correspondence between constraints and items; and for the nature of the data used to make the estimations, we use a MULTILOG input matrix with the same meaning and format. That is, a true value, indicating the knowledge of certain concept; and a false value, in those cases in which this concept is erroneously known.

Finally, we should mention that MULTILOG was used in both cases, in order to ensure that calibration and inference techniques were the same.

6.3 Results

To carry out the comparison of both estimations, a paired t-Student was used with 95% confidence factor. This statistic is commonly used to compare the difference between two small-sized populations. The null hypothesis was that the difference between the population means was zero. The analysis gave a p-value of 0.7972 which clearly suggests that we cannot reject the previous null hypothesis and, therefore, there was no significant difference between the evaluations obtained with OOPS, using our model which combines IRT and CBM, and those given by Siette where the 3PL model was applied.

7 Conclusions and Future Work

The techniques described in this paper provide several advantages to the procedural activities assessment. The student knowledge estimations are invariant, i.e. they do not depend on the set of problems solved by the student, and the estimation accuracy degree can be controlled. Furthermore, it is a statistical inference mechanism where CCCs are estimated using historical information about a sample of the student performance while they took these problems.

Our initial hypothesis was that the most popular statistical techniques used for determining the student knowledge in testing, can also be used in procedural activities. Therefore, through these procedural activities, we can obtain similar results to those obtained with an IRT-based test. The conclusions of our experiment with ill-defined tasks, suggest that our hypothesis is correct. As a consequence, we can carry out well-founded assessments, using only a few procedural activities. To obtain a similar assessment result through tests, a high number of questions would be needed.

IRT-based tests can be administered adaptively, through *adaptive testing*, where each question is selected dynamically, in terms of the student estimated knowledge level. Moreover, the test size is also decided dynamically in terms of the required estimation accuracy. In this sense, our future work will focus on the development of a mechanism for the adaptive administration of procedural activities.

In this work, we have estimated the student declarative knowledge through procedural activities. Currently, we are working on extending the model to also evaluate procedural knowledge. Likewise, since the model presented in this paper has been applied to two different domains, it is also necessary to extend the study of another evaluation system which confirms the hypothesis stated in these experiments.

Acknowledgments. This work has been co-financed by the Spanish Ministry of Science and Innovation (TIN2007-67515) and by the Andalusian Regional Ministry of Science, Innovation and Enterprise (TIC-03243). The authors wish to thank prof. David Bueno and his students for their willingness to participate in the experiments described in this paper.

References

1. Brooks, R.A.: Intelligence without representation. Artificial Intelligence 47, 139–159 (1991)
2. Greer, J., McCalla, G.: Student Modeling: The Key to Individualized Knowledge-based Instruction. Springer, Heidelberg (1994)
3. Thurstone, L.L.: A method of scaling psychological and educational tests. Journal of Educational Psychology 16, 433–451 (1925)
4. Anderson, J.R., Boyle, C.F., Corbett, A.T., Lewis, M.W.: Cognitive modeling and intelligent tutoring. Artificial Intelligence 42, 7–49 (1990)
5. Conati, C., Gertner, A., VanLehn, K.: Using Bayesian networks to manage uncertainly in student modeling. User Modeling & User-Adapted Interaction 12(4), 371–417 (2002)
6. Hayes, J.R.: Cognitive psychology: Thinking and creating. Dorsey Press, Homewood (1978)
7. Gálvez, J., Guzmán, E., Conejo, R., Millán, E.: Student Knowledge Diagnosis Using Item Response Theory and Constraint-Based Modeling. In: The 14th International Conference on Artificial Intelligence in Education (AIED 2009), vol. 200, pp. 291–298 (2009)
8. Ohlsson, S.: Constraint-based Student Modeling. In: Student Modeling: The Key to Individualized Knowledge-based Instruction, pp. 167–189. Springer, Heidelberg (1994)
9. Mitrovic, A., Martin, B., Suraweera, P.: Intelligent Tutors for All: The Constraint-Based Approach. IEEE Intelligent Systems, IEEE Educational Activities Department 22, 38–45 (2007)
10. Ohlsson, S., Mitrovic, A.: Constraint-based knowledge representation for individualized instruction. Computer Science and Information Systems 3, 1–22 (2006)
11. Mayo, M., Mitrovic, A.: Optimising ITS behaviour with Bayesian networks and decision theory. International Journal of Artificial Intelligence in Education 12, 124–153 (2001)
12. Hambleton, R.K., Swaminathan, H., Rogers, J.H.: Fundamentals of Item Response Theory (Measurement Methods for the Social Science). Sage Publications, Inc., Thousand Oaks (1991)
13. Guzmán, E., Conejo, R., Pérez-de-la-Cruz, J.L.: Adaptive Testing for Hierarchical Student Models. User Modeling and User-Adapted Interaction 17, 119–157 (2007)
14. Gálvez, J., Guzmán, E., Conejo, R.: HA blended E-learning experience in a course of object oriented programming fundamentals. Knowledge-Based Systems 22(4), 279–286 (2009)
15. Gálvez, J., Guzmán, E., Conejo, R.: A SOA-Based Framework for Constructing Problem Solving Environments. In: The 8th IEEE International Conference on Advanced Learning Technologies, pp. 126–128 (2008)
16. Friedman-Hill, E.J.: JESS, The Java Expert System Shell. SAND–98-8206 (1997)
17. Guzmán, E., Conejo, R., Pérez-de-la-Cruz, J.L.: Improving Student Performance using Self-Assessment Tests. IEEE Intelligent Systems 22, 46–52 (2007)
18. Thissen, D.: Multilog: Multiple, categorical item analysis and test scoring using item response theory (version 5.1). Mooresville, In Scientific Software (1988)