Exploring Quality of Constraints for Assessment in Problem Solving Environments

Jaime Galvez Cordero, Eduardo Guzman De Los Riscos, and Ricardo Conejo Muñoz

Universidad de Malaga,
29071, Malaga, Spain
{jgalvez,guzman,conejo}@lcc.uma.es

Abstract. One of the approaches that has demonstrated by far its efficiency as a tutorial strategy in problem solving learning environments is the Constraint-Based Modeling (CBM). In existing works it has been combined with a data-driven technique for automatic assessment, the Item Response Theory (IRT). The result is a well-founded model for assessing students while solving problems. In this paper a novel technique for studying quality of constraints for this type of assessment is presented. It has been tested with two new systems, an independent component for assessment that implements CBM with IRT, which provides assessment to a new problem solving environment developed to assess the students' skills in decision-making in project investments. The results of testing our approach and the application of these two systems with undergraduate students are also discussed in this paper.

Keywords: Problem Solving Environments, Constraint-Based Modeling, Item Response Theory.

1 Introduction

Among the existing approaches that can be applied to modeling students in problem solving environments, Constraint-Based Modeling (CBM) has proved its effectiveness with a range of tutors and studies performed in the last years [1]. It is easier to be applied than other approaches, such as Model Tracing [2], since CBM does not require identifying all possible steps a student could take to reach a solution to a problem. On the contrary, only those constraints that any solution should not violate need to be identified.

CBM is an effective approach, whose power lies in the design of the constraints set. To build a new learning environment using authoring tools such as ASPIRE [3] is a very easy task, since no programming skills are needed. What is necessary to model constraints in an appropriate manner is to have a broad knowledge of the domain matter; the same happens in any other approach when a new learning environment is going to be developed. Nevertheless, even with human experts, constraints could not be reflecting properly a domain principle. In this sense, a constraint could actually represent a more specific principle or, otherwise, a more general one.

S.A. Cerri et al. (Eds.): ITS 2012, LNCS 7315, pp. 310–319, 2012.

The work presented here is based on the model presented in [4, 5] which combines Item Response Theory (IRT) with CBM. IRT is a data-driven theory commonly used in testing environments for assessment. The IRT+CBM model generates probabilistic curves, called Constraint Characteristic Curve (CCC), which are inferred from a calibration process with prior data from students' performance.

Unfortunately, as mentioned before, constraints may not represent the domain model in the best possible way. Moreover, the calibration performed by the IRT+CBM model might not have enough evidence to infer the CCCs properly. In this paper we present a data-driven technique to determine quality of constraints, i.e., whether or not they are good enough to be used for assessment.

The content of the article is organized as follows: first, the work related to our research is mentioned. Then, we describe how IRT would help to determine quality of constraints. Next, we present a new assessment framework and a new problem solving environment we have used to carry out the experimentation. Section 5 describes our hypothesis, the experiment we designed, and our findings. Finally, conclusions and future research work are outlined in the section 6.

2 Related Work

The first methodology of interest to the work of this paper is the CBM, which is used to model the domain and student in problem solving environments with the goal of improving learning of a given subject. Its basis is the Olsson's theory of learning from performance errors [6], according to which incomplete or incorrect student's knowledge can be used within an intelligent tutoring system as guidance. Detection of this faulty knowledge is done by the main element of CBM: the constraint, which represents a principle that none of the possible solutions to a problem in this domain will violate.

The other technique employed here is the IRT conceived by Thurstone [7], a well-founded theory used in testing environments to measure certain traits, such as the student's knowledge. This theory is based on modeling the probability of answering a question/item correctly given a student's knowledge level by means of a function called Item Characteristic Curve (ICC) where the greater the student's knowledge level is, the higher the probability of answering correctly.

The main work related to the study conducted here is based on [4, 5], where a model combining CBM and IRT is proposed in order to provide CBM with a long-term student model. According to this work, constraints of CBM are equivalent to questions of a test and using IR assessment over constraints can improve the student model accuracy and, consequently, provide a better adaptation to the student learning process. The analogy made between these two methodologies is the basis that allowed us to apply techniques associated with the IRT into CBM to develop this work.

In literature there are works on CBM [8, 9] which explore whether or not groups of constraints, linked to more general concepts, would be more effective for learning than single constraints. However, our approach treats it from a different point of view since it is based on IRT.

3 Using IRT to Study Quality of Constraints for Assessment

The analogy that allows us to formulate the approach explained below is that constraints are equivalent to questions in the sense that both of them represent declarative knowledge units and both of them have two values as the result of the student performance: one positive and one negative. The positive value represents correct knowledge, which, in the case of CBM, corresponds to a satisfaction of a constraint and, in questions, to a correct response. The negative value would represent faulty knowledge, meaning that the constraint was violated or the response was wrong.

According to [4, 5], to apply IRT to constraints, a Constraint Characteristic Curve (CCC) is defined for every constraint in a calibration process with the evidence taken from the student's performance. As in IRT, it represents a probability distribution based on the knowledge: the broader the knowledge, the more probability of satisfying the constraint. Violations can be also modeled using the inverse of this function, which means that when the knowledge is broader, the probability of violation is lower. As a result of the calibration, the parameters representing the CCC are obtained.

Normally, the 3 parameters logistic function (3PL) is applied, producing the following three parameters: *a* represents discrimination which is a value proportional to the slope of the curve. The higher it is, the greater capacity to differentiate between the students' inferior and superior knowledge levels; *b* is the difficulty and it corresponds to the knowledge value for which the probability of satisfying the constraint is the same as that of violating it; the last parameter, *c*, is the guessing and it represents the probability that a student will satisfy the constraint even though he/she may not possess the knowledge required to do so.

The basis of our proposal is that, considering the parameters of a CCC, we could manage constraints as if they were items and, consequently, mechanisms applied over items to determine their quality are equally valid for constraints. Concretely, we propose to employ the Item Information Function (IIF) [10, 11], which is a technique used in adaptive testing in order to describe, select, and compare items and tests. Accordingly, we define the Constraint Information Function (CIF) that can be used to detect the most suitable constraints for assessment (see equation 1 based on [10]). In this way, assessment would be done over concepts representing more faithfully the reality, which would reduce misleading result of an inappropriate representation.

$$\mathrm{I_i}(\theta) = \frac{2.89a_i^2(1-c_i)}{\left[c_i+e^{1.7a_i(\theta-b_i)}\right]\left[1+e^{-1.7a_i(\theta-b_i)}\right]^2} \quad (1)$$

The $\mathrm{I_i}(\theta)$ represents how informative a constraint *i* is for a fixed value of the student's knowledge, θ. This knowledge ranges from $-\infty$ to ∞, but in practice, only values from the interval [-4, 4] or [-3, 3] are normally considered because, out of this interval, the value of the CIF is very close to zero and hence it is negligible. Within this interval the function has a logistic bell shape with values close to zero in the extremes and a maximum in the value of $\theta=b_i$, which is the parameter corresponding to the difficulty of the constraint and the most representative for the CIF. Note that equation 1 assumes that CCC has been calibrated under the 3PL model.

To calculate the CIF of a particular constraint, given that the formula is the derivative respect to θ, we would apply equation 2 to get the total information, which would consider the whole range of student's knowledge.

$$I_i = \int_{-\infty}^{\infty} I_i(\theta)\ d\theta \quad (2)$$

We distinguish three particular cases where CIFs could help to explore the quality of constraints:

a) The first case is related to the relevance of constraints. Some of the domain constraints are not always relevant to all the problems. They will have less evidence in comparison to others and, thereby, less information of the domain. The use of these constraints to assess students could produce an inaccurate assessment.
b) Secondly, extremely high values of the information function in a constraint, in comparison to the others, could suggest that this constraint is grouping more than one domain principles. The recommendation here should be to consider splitting this constraint into several ones, each one modeling a more specific principle.
c) The last case would be exactly the opposite of the second one: the value of the information function is extremely low. Two reasons could lead to this fact: first, the population is small and there is not enough evidence to calibrate the curves properly; and second, the constraint is too fine-grained and it should be merged with other constraint to model a more general principle. Finally, this CIF value could also suggest that the constraint is not a good indicator of the student's knowledge in the domain.

Regarding the distinction between good and bad constraints, it is clear that if the information is lower, it will be worse for assessment. Nevertheless, if we have to establish a limit or threshold to separate good constraints from bad ones, we still do not know if there is a common limit for different domains. In the experimentation section we give the threshold, obtained for our problem solving environment, as a reference point for further studies.

4 Tools Used in the Experiment

To perform the experiment we used three systems, each one for different purposes: the first one is Siette [12], a web-based authoring tool and testing environment where students can take tests on a subject matter, and where assessment with IRT is possible. The other two systems are presented in this paper for the first time and both are components of a bigger platform for teaching mathematics, DEDALO [13]. Following the philosophy of this framework, every component is independent and can communicate through Web Services with the rest of the platform components. These components are called Project Investments Problem Solving Environment (PIPSE) and CBM-Engine.

4.1 Project Investment Problem Solving Environment

PIPSE was developed to be used as part of a course of Project Management as a support tool. It is a problem solving environment focused on the study of the profitability of starting up a project given a series of variables associated with costs and benefits that it would generate. The system is a Web application implemented on *.net* through which students can apply several indexes, such as Net Present Value (NPV) or Internal Rate of Return (IRR) [14], to study the profitability of a project. Figure 1 shows the four main parts of PIPSE: A is a panel of actions related to the current session and to the student's attempts; B contains the problem stem and buttons to hide / show it; C is the table with the student's solutions which can be edited; and D contains the controls to add years or variables to the problem, with the solution variables and a workspace panel where all actions carried out by the student are represented, and new commands can be entered into a command line interpreter.

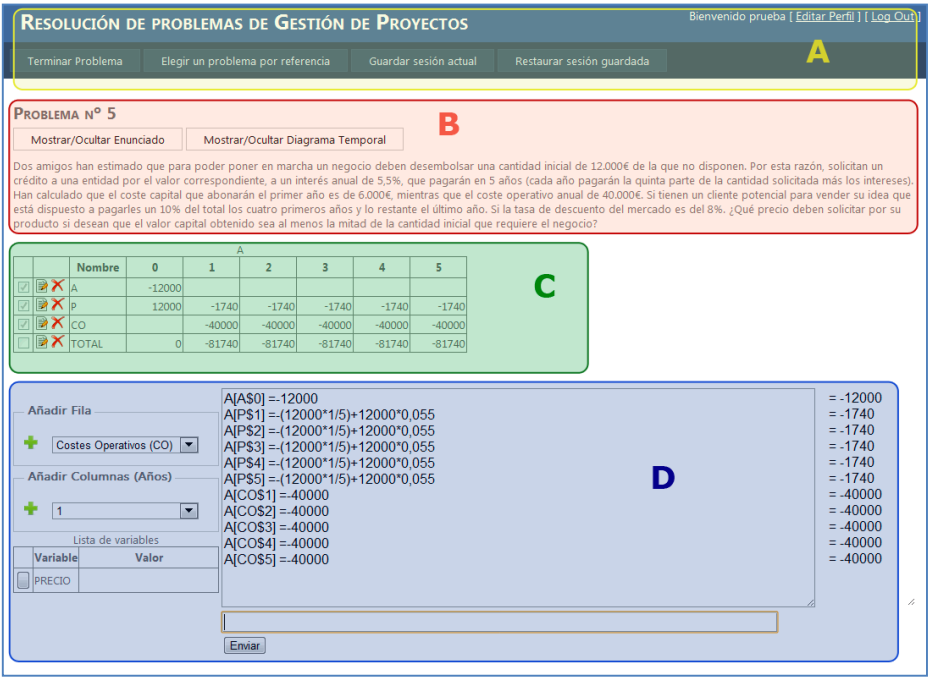

Fig. 1. Project Investment Problem Solving Environment

The system interface tries to reduce the cognitive overload [15], otherwise calculus inherent in this kind of problems would affect the student's working memory. This is done by providing students with mechanisms similar to a datasheet, allowing them to use references to cells of a table to build formulas that will be automatically interpreted and calculated by the system. Those mechanisms make calculations unnecessary outside the interface and help students to focus on using their knowledge to solve the problem. Students should build a table with all the problem information and

provide other information, which, all together, would represent the solution to the problem. PIPSE is able to present information about the student performance errors obtained from the application of CBM to their solution. This characteristic makes the system not only an assessment tool, but also, suitable for learning purposes.

Information gathered from the student interaction with the system is used by it to generate different assessments. To accomplish this, the information is sent to different assessment subsystems, available through Web Services. Those subsystems are independent and they are not fixed, i.e., they can be dynamically replaced, added, or removed from the system. Although currently there are two different assessment subsystems implemented, each one associated with a different methodology, only one of them is of interest to this study: the one that implements the combination of CBM+IRT, which is explained in the following subsection.

4.2 CBM-Engine Assessment Component

The CBM-Engine is a SOA-based component following the same idea of [16] that implements CBM with IRT assessment. It has no interface but a set of services that can be used to apply the already-explained methodology in any external system/tutor. It is formed by a three-layered architecture comprising: a) a top level layer offering Web Services as interface with the external systems, b) an assessment layer where all inferences and application logic are carried out, and c) a persistence layer in charge of storing data structures common to any domain and those specific to each particular domain. New problem solving environments or tutors wanting to obtain assessment with this framework must be added to the system by using an authoring tool where constraints and data structures must be defined.

In the particular case of the PIPSE system, we are dealing with a well-defined domain where problems as well as tasks are well-defined [17]. The constraints and the specific data structures forming the domain model were added to the engine resulting in a set of 17 constraints, which can be categorized in three subsets: (a) correct definition of variables related to the problem; (b) manipulation of the data in the solution table; and (c) calculus and inference associated with the solution.

5 Experimentation

In this section we are going to describe the experiment we have conducted to validate our proposal. In this sense, the main hypothesis to be tested will be whether or not the IIF can be applied to constraints in the same way it is used in testing environments, to detect constraints not suitable for assessment.

As a secondary goal, the second part of our experimentation tries to study an important characteristic that any system should have in order to be used for assessment purposes: it should be able to provide a valid assessment of the student performance. To verify this with the PIPSE system presented in this study, we proceeded as it was done in [4, 5]. Following the same criteria, assessment produced with the system using the combination CBM+IRT should be similar to the one obtained by applying a

formal assessment of the same concepts involved in the system. Thus, the second part is focused on exploring whether or not the assessment provided by our new system, using a set of constraints valid for assessment, is equivalent to the one provided by a test where IRT would be applied to infer the student's knowledge.

5.1 Design and Implementation of the Experiment

In order to evaluate our methodology, we designed an experiment with students from the last year of the M.Sc. in Computer Science degree at the University of Malaga. A total of 24 students participated in the study that was performed in December 2011 and comprised of several stages. First, the students were instructed during several classes on the different indexes to solve the project investment problems. Next, they took a one-hour-long session where they were able to use the system to solve two problems seen previously in class; a week later, we performed a paper-based exam where two problems where proposed and a test was administered.

To test the experiment hypothesis, problems proposed in the exam did not cover the whole set of constraints; a characteristic we would use later in the analysis of constraints quality with the CIF. Regarding the test, it was designed, following the same premises as in [4, 5], in order to assess the same concepts involved in the problems. To achieve this, a question was written for each constraint, producing a total of 15 questions in the test. Two of the constraints were left out of the test since they were not associated with concepts, but with mathematical verifications.

Unlike the early work with this technique, the exam was made on paper with the aim of getting only the constraints violations and preventing students from receiving any type of feedback. With this omission of information about errors made in the solution, the learning factor associated with feedback was isolated and taken out of the experiment, which, according to IRT requirements, is important to generate a good calibration of constraints and to apply IRT mechanisms. Once all the students had finished the exam, the solutions they provided were then introduced into the problem solving environment and constraints were checked against them.

The experiment was used as an assessment item in the course, and all 24 students enrolled in the course participated in it. Additionally, the Siette test was also administered to the students. After all data had been gathered from students, we performed the analysis of constraints applying the approach explained before, filtering some of the constraints and leaving the rest to perform the assessment of students, which led us to the results described in the next section.

5.2 Results

The solution provided by every student was introduced into the PIPSE, which sent it to the CBM-Engine, recording all data and calibrating constraints. The calibration output, i.e., parameters representing the CCC, was analyzed by applying the information function to every constraint using the formula (1). As a result, we got an average value of 14.81 of the CIF and a standard deviation of 2.18 for the whole set of 17 constraints.

Before examining the results, we grouped the constraints into those that were not relevant during the problems taken in the exam and those that were. Looking at the results, the first supporting finding we made was that the group of relevant constraints, composed by 7 of them, had a greater mean of the CIF (16.29 versus 13.76). Although after a t-test we couldn't find significant difference in their means (p-value 0.68), we discovered that one of the constraints from this analysis had a strange value that was affecting the results by introducing noise. When we discarded it, the difference became significant (p-value 0.012).

Besides, we ordered the constraints according to their value in the CIF, finding that 5 out of 7 of the relevant constraints were at the top of the list. In this particular case, splitting the data with the threshold $\bar{x} + 0.5\sigma$, resulted in the division of the relevant constraints at the top of the list. This suggests that most of them could be detected using the CIF (conforming case *a)* of our proposal in section 3). Regarding the other two relevant constraints not found at the top, both of them were at the bottom with an order of -1.67 times the standard deviation, which was significant. This constraint with extremely low value was representing a principle of the domain that was implicit in other constraints and, therefore, it was not providing much information. The other constraint at the bottom of the list was not significantly different from the rest and experts in the domain didn't find any other constraint that could be merged with it. This probably is explained by a small population of students that didn't provide enough evidence to get a good calibration of the constraint. In any case, irregularities of both of these constraints were detected with our approach (conforming case *c)* of our proposal).

Additionally, during the analysis we found a constraint with an outstanding value of the information function over the remaining ones. It had a 20.07, which is an order of 2.4 times the standard deviation. Since we had not deliberately designed this constraint to be different from the others, by examining it to see what the cause of this exaggerated value would be, we realized it was due to grouping several concepts together, which led to students' faulty knowledge being more pronounced here. It means that we were able to detect a constraint which could be split into others representing more fine-grained principles of the domain (see case *b)* of our proposal in section 3).

The filtered set of constraints was used then in the assessment framework to provide a score for every student. This assessment was compared with the one obtained in the Siette test using a paired t-test at 95% confidence. As result of the t-test we got a p-value of 0.8155. This clearly suggests that in the case of pairs of scores belonging to a student, there is no significant difference between them. Furthermore, we performed a correlation analysis between both scores, obtaining a correlation coefficient of 0.06. This is a very small value that we think could be explained by two factors: a) the number of data from students / constraints is not big enough; or b) questions of the test were not correctly designed to evaluate the same concepts.

6 Conclusions

In this paper, a new approach, called Constraint Information Function (CIF), to study quality of constraints in CBM tutors has been introduced. This methodology is based

on the analogy discovered between questions and CBM constraints [4, 5], according to which, constraints are used as if they were questions in a test and, consequently, mechanisms of IRT can be applied to constraints. In this way, the IIF, normally used to study quality of questions in test development, has been proposed to determine whether or not constraints are representing the domain correctly and if they can be used for assessing students appropriately. This approach would help to generate a more accurate assessment, leading to a more precise student model and a better adaptation. In addition, our approach could contribute to the constraint elicitation process, since it could help to detect constraints that should be split or grouped, and even to reformulate or discard them.

As part of the study, the CBM with IRT assessment has been implemented in a new SOA-based assessment framework called CBM-Engine. This system is able to perform the same assessment procedure combining both techniques, with the advantage of being independent of the learning system. What is more, it can be used by any external learning environment as long as it is registered in the system and its domain model is incorporated into the specific domain data structures.

Besides, a new problem solving environment focused on the domain of project investment analysis has been presented. It has been designed to provide different assessments from independent subsystems, each one using different assessment mechanisms. For the study presented in this article, only the methodology provided by the CBM-Engine is of relevance. This problem solving environment can be used not only as an assessment tool, but also as a tutoring system since it is able to take the feedback produced by the CBM and present it to the students. However, this scaffolding mechanism goes beyond the scope of this paper.

In the experiments conducted, we used the problem solving environment working with the new assessment framework. Students' data were used by the framework to produce first a calibration of constraints and then an assessment. Between the two phases, the Information Function was successfully applied to detect those constraints which were not suitable to be used for assessment. The assessment performed after filtering the non-suitable set of constraints was compared to the assessment of a test covering the same concepts involved in the constraints. Statistical analysis suggests that our model could diagnose in the same way as an IRT-based test does. Nevertheless, no much correlation was found between the test and the problem solving scores, probably because the data used in the experiment was much reduced.

When we look at CBM with IRT as a problem solving environment assessment mechanism, the results are promising and a range of possibilities is opened with this synergy. Nevertheless it has a drawback that should be taken under consideration: so far, results have been found only in systems without a big population using it. Therefore, further work is being done to explore efficiency of this technique for bigger systems. Further work should be also done to explore if the process of the approach presented here, which was made entirely manual, could be automated within the CBM-engine; if some common threshold to distinguish good constraints from bad ones can be found in different systems; and whether there exist any automatic mechanism to determine it. Our current work is focused on these lines and exploring other utilities of IRT mechanisms that can be applied to CBM tutors.

Acknowledgements. This work has been co-financed by the Andalusian Regional Ministry of Science, Innovation and Enterprise (P07-TIC-03243 and P09-TIC-5105).

References

1. Mitrovic, A., Martin, B., Suraweera, P.: Intelligent Tutors for All: The Constraint-Based Approach. IEEE Intelligent Systems 22, 38–45 (2007)
2. Mitrovic, A., Koedinger, K.R., Martin, B.: A comparative analysis of cognitive tutoring and constraint-based modeling. In: User Modeling, pp. 313–322 (2003)
3. Mitrović, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., Holland, J.: Authoring Constraint-Based Tutors in ASPIRE. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 41–50. Springer, Heidelberg (2006)
4. 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, vol. 200, pp. 291–298 (2009)
5. Gálvez, J., Guzmán, E., Conejo, R.: Data-Driven Student Knowledge Assessment through Ill-Defined Procedural Tasks. In: Meseguer, P., Mandow, L., Gasca, R.M. (eds.) CAEPIA 2009. LNCS(LNAI), vol. 5988, pp. 233–241. Springer, Heidelberg (2010)
6. Ohlsson, S.: Constraint-based student modeling. In: Student Modeling: the Key to Individualized Knowledge-based Instruction, pp. 167–189 (1994)
7. Thurstone, L.L.: A method of scaling psychological and educational tests. Journal of Educational Psychology 16, 433–451 (1925)
8. Martin, B., Mitrović, A.: Using Learning Curves to Mine Student Models. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 79–88. Springer, Heidelberg (2005)
9. Martin, B., Mitrović, A.: The Effect of Adapting Feedback Generality in ITS. In: Wade, V.P., Ashman, H., Smyth, B. (eds.) AH 2006. LNCS, vol. 4018, pp. 192–202. Springer, Heidelberg (2006)
10. Birnbaum, A.: Some latent trait models and their use in inferring an examinee's ability. In: Statistical Theories of Mental Test Scores. Addison-Wesley, Reading (1968)
11. Hambleton, R.K., Swaminathan, H., Rogers, H.J.: Fundamentals of Item Response Theory. Sage Publications, Inc., Thousand Oaks (1991)
12. Guzmán, E., Conejo, R., Pérez-de-la-Cruz, J.L.: Improving Student Performance using Self-Assessment Tests. IEEE Intelligent Systems 22, 46–52 (2007)
13. DEDALO project, Assessment and Learning of Mathematics (January 23, 2012), `http://dedalo.lcc.uma.es`
14. Khan, M.Y.: Theory & Problems in Financial Management. McGraw Hill Higher Education, Boston (1993)
15. Sweller, J., van Merrienboer, J., Pass, F.: Cognitive architecture and instructional design. Educational Psychology Review 10(3), 251–296 (1998)
16. Gálvez, J., Guzmán, E., Conejo, R.: A SOA-Based Framework for Constructing Problem Solving Environments. In: ICALT 2008, pp. 126–127 (2008)
17. Mitrovic, A., Weerasinghe, A.: Revisiting ill-definedness and the consequences for ITSs. In: 14th International Conference on Artificial Intelligence in Education, pp. 375–382 (2009)