Int J Artif Intell Educ (2013) 23:22–49
DOI 10.1007/s40593-013-0008-2

RESEARCH ARTICLE

Statistical Techniques to Explore the Quality of Constraints in Constraint-Based Modeling Environments

Jaime Gálvez · Ricardo Conejo · Eduardo Guzmán

Published online: 6 November 2013

Abstract One of the most popular student modeling approaches is Constraint-Based Modeling (CBM). It is an efficient approach that can be easily applied inside an Intelligent Tutoring System (ITS). Even with these characteristics, building new ITSs requires carefully designing the domain model to be taught because different sources of errors could affect the efficiency of the system. In this paper a novel mechanism for studying the quality of the elements in the domain model of CBM systems is presented. This mechanism combines CBM with the Item Response Theory (IRT), a data-driven technique for automatic assessment. The goal is to improve the quality of the elements that are used in problem solving environments for assessment or instruction. In this paper we propose a set of statistical techniques, i.e., the analysis of the point-biserial correlation, the Cronbach's alpha and the information function, to explore the quality of constraints. Two different tools have been used to test this approach: a problem solving environment designed to assess students in project investment analysis; and an independent component that performs assessments using CBM and IRT. Results suggest that the three methods produce consistent diagnosis and may be complementary in some cases. In the experiments we have carried out they were able to detect faulty, bad and good quality constraints.

Keywords Problem solving environments · Constraint-based modeling · Item response theory

Introduction

Among the existing approaches that can be applied to modeling students in problem solving learning environments, Constraint-Based Modeling (CBM) has proved its

J. Gálvez (✉) · R. Conejo · E. Guzmán
Universidad de Málaga, 29071 Málaga, Spain
e-mail: jgalvez@lcc.uma.es

R. Conejo
e-mail: conejo@lcc.uma.es

E. Guzmán
e-mail: guzman@lcc.uma.es

effectiveness with a range of tutors and studies performed in the last few years (Mitrovic 2012; Mitrovic et al. 2007). It is easier to apply than other approaches, such as Model Tracing (Mitrovic et al. 2003) since CBM does not require the identification of all possible steps a student could take to reach a solution to a problem. On the contrary, only those principles (called *constraints*) that no solution should violate need to be identified.

CBM is an effective paradigm, the power of which lies in the design of the constraints set. This set is the most important element of this modeling paradigm in order to conduct intelligent tutorial actions. Designing the constraints set in new learning environments can be performed very easily using authoring tools such as ASPIRE (Mitrovic et al. 2006) as no programming skills are needed. What is necessary to appropriately model constraints is a broad knowledge of the domain matter; the same as with any other approach when a new learning environment is going to be developed.

Despite the instructional efficiency of CBM and the reduced complexity required to design the knowledge base with respect to other approaches, even if domain experts are in charge of this task, resulting constraints may not properly reflect a domain principle. The human factor and other sources of error could mean that the constraints fail to correctly represent what they were intended to and, therefore, instruction and other educational methods relying on CBM could be misleading or inaccurate.

The work presented here is based on the model presented in (Gálvez et al. 2009a, b, 2012) which combines Item Response Theory (IRT) with CBM. IRT is a data-driven theory commonly used in testing environments for assessment, i.e., to produce a quantitative judgement about some subject's unobservable trait. The IRT + CBM model generates probabilistic curves that are inferred from a calibration process with prior data from students' performance. This approach enables assessments of problem solving using the evidence collected by an Intelligent Tutoring System (ITS).

The goal of this research is to apply well-founded mechanisms borrowed from IRT into CBM tutors in order to a) make these systems more reliable and b) to extend assessment mechanisms to problem solving environments. The approach presented in this paper is the application of one of three mechanisms, in this case to statistically study the quality of constraints for assessment purposes.

The content of the article is organized as follows: first, the background concepts associated with this work are provided. Second, the work related to our research is shown. Third, we describe how three different statistical techniques would help to determine the quality of the constraints. Then, we present the new assessment framework and the problem solving environment we have used to conduct the experimentation. Next, our hypothesis, the experiment we designed, and our findings are discussed. Finally, conclusions and future research work are outlined.

Background

Constraint-Based Modeling

The first methodology of interest to the work of this paper is the CBM, which defines a paradigm to model the domain and student in problem solving environments in

order to improve learning in a given subject. Since this approach first appeared, many tools and studies have been conducted to test its validity for instruction. As collected in (Mitrovic 2012; Mitrovic et al. 2007), results obtained during the last 16 years from research in a number of areas within ITSs (affect, collaboration, open learner modeling, tutorial dialogs, etc.) reflect its evident success.

The lead in the research and advances associated with this approach has been taken by researchers of the Intelligent Computer Tutoring Group at the University of Canterbury (New Zealand). Mitrovic (2012) describes a long-term research line using CBM Tutors, starting with SQL-Tutor, a tutor for select queries in databases; followed by other tutors in the database domain such as NORMIT and EER-Tutor; a number of diverse tutors in other domains; and even an authoring tool, such as ASPIRE. The success of these systems is also a proof of the important of this technique.

The basis of CBM is Ohlsson's theory of learning from performance errors (Ohlsson 1994, 1996), according to which learning occurs in two phases: first, incomplete or incorrect student's knowledge is detected through problem solving and, later, it can be used within an ITS as part of instructional strategies. In more detail, the characteristics of these two phases are:

- Detection of faulty knowledge is done by the main element of CBM: the constraint, which represents a principle that must be followed by all correct solutions to a problem in a particular domain. Irrespective of the sequence of steps performed to solve a problem, faulty knowledge will produce solutions with wrong elements and, therefore, principles will be violated. The entire set of constraints encoding the domain principles comprises the domain model, which is completed with the set of problems that can be presented.
- Remediation in CBM takes place firstly by showing immediate feedback to the students when a constraint is violated. In this way, they are warned that the knowledge is incorrect, encouraging them to correct it. Second, the learning process is guided by adapting the next problem to be tackled, by selecting the one involving the problematic domain principles. Both of them are driven by the content of the student model, which is comprised of the students' performance outcomes, i.e., the list of violated and satisfied constraints. Moreover, estimates of the student knowledge level are introduced as part of the student model to drive instructional strategies.

Although the original formulation of CBM did not impose any restrictions upon how constraints should be encoded they are normally implemented as inference rules with an antecedent and a consequent. While the first contains logical conditions to detect the violation of a principle, i.e., the faulty knowledge; the second contains internal procedure calls associated with the instructional actions to be performed when these situations are detected. Constraints are evaluated with respect to the state of a solution being built in a given problem, and only when the antecedent has been fired due to a violation, is the second element executed.

The antecedent of this typical implementation is modelled as a pair of conditions: a *relevance condition*, which establishes when a constraint is applicable; and a *satisfaction condition*, which defines the required tests to check the correctness of the solution. When the relevance condition is satisfied in a solution, it is said that it is d for this solution and the associated problem. Only then, is the satisfaction condition checked and the result of satisfying or violating a constraint is used by the CBM system.

The correct solution of a problem could make a set of constraints relevant. However, students could have omitted parts of that solution, which would prevent some constraints from becoming relevant. Therefore, in a data-driven approach such as the one that will be presented further on in this paper, the relevance of a constraint is strongly related to the solutions submitted to the system. Taking that into account, from now on, we will refer to non-relevant constraints as those with no evidence about the violation or satisfaction thereof, and for any student.

Although CBM establishes how to model the domain and the student in an easy and efficient way, it suffers from a major drawback which its authors pointed out in (Ohlsson and Mitrovic 2006): it is necessary to have a long-term student model in order to improve the pedagogical actions a tutor could take. Nowadays, most of the approaches implemented use heuristics to model the probability of having learnt a constraint. Probabilistic methods based on Bayesian networks have also been implemented (Mayo and Mitrovic 2001; Mitrovic et al. 2002). They require a considerable knowledge engineering effort in each new system development. Mayo and Mitrovic (2001) declare that "One area of concern with this approach is scalability, both to larger domains and different domains". One possible way to solve this could be applying Bayesian Knowledge Tracing (BKT) to constraints. We believe this is possible since both key elements, constraints and rules of model tracing are modeled using inference rules. In this paper we present an alternative approach that, based on IRT techniques, also overcomes this issue (Gálvez et al. 2009a, b).

Item Response Theory

IRT was conceived by Thurstone (1925) as a theory used in testing environments to measure certain latent traits, such as the student's knowledge. This theory is based on two main principles (Hambleton et al. 1991): (1) The answer of the students to a given question (commonly called *item* in the literature), can be explained according to their knowledge level, which can be measured as an unknown numeric value θ; and (2) the conditional probability of answering an item correctly given a student's knowledge level, can be modelled by means of a function called the Item Characteristic Curve (ICC).

In the ICC, the greater the student's knowledge level, the higher the probability of answering correctly (see Fig. 1). An item curve can be represented using a number of different models (DeMars 2010). Nevertheless, parametric ones are normally applied due to their simplicity since they use a reduced set of parameters instead of the whole range of values of the curve. In particular, one of the most popular is the 3-Parameter-Logistic-Function (3PL) which uses three parameters, each with different meanings. In 3PL the probability $P_i(\theta)$ of correctly answering an item i is described by the equation:

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

where:

- The parameter a_i represents discrimination, which is a value proportional to the slope of the curve. The higher its value, the greater the capability to differentiate

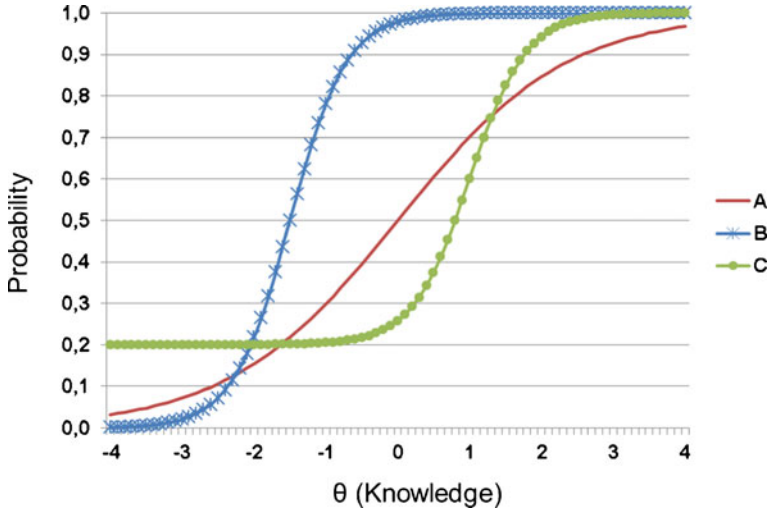


Fig. 1 Different ICCs

between the students with higher and lower knowledge levels. For example, the ICC labelled *A* in Fig. 1 has less discrimination than *B* since its slope is lower.

- b_i is the difficulty and it corresponds to the knowledge value for which the probability of answering correctly the item is the same as failing it. The value of this parameter in the example of Fig. 1 is 0 for curve *A*, -1.5 for curve *B*, and 1 for curve *C*.
- The last parameter c_i, is called the guessing factor and it represents the probability that a student will answer the item correctly even though he/she may not possess the knowledge required to do so. The guessing parameter can be distinguished in curve *C* of Fig. 1 with a value of 0.2 and 0 in the other curves.

There are different response models in IRT. The simplest dichotomous model assumes that the answer u_{ij} of a student j to a given item i is defined as *1* if the student gives the correct answer and *0* otherwise. The knowledge level θ_j of each student, and the item parameters a_i, b_i, and c_i, can be estimated using different methods. This estimation process is known as *calibration* and tries to adjust the parameter values to fit the performance observed in the sample. One of the most popular is the Maximum Marginal Likelihood (MML), which tries to find out the values where the likelihood function $L(u|\theta)$ has a local maximum:

$$L(u|\theta) = L(u_1 \ldots u_n|\theta) = \prod_{i=1}^{n} P_i(\theta)^{u_i}(1-P_i(\theta))^{(1-u_i)} \left(\frac{\partial L(u|\theta)}{\partial \theta}\right)_{\theta=\widehat{\theta}} = 0 \qquad (2)$$

Another useful function in IRT is the *Item Information Function* (*IIF*) (Birnbaum 1968; Hambleton et al. 1991), which tells us about the contribution of a single item to our understanding of the latent trait. The *IIF* tries to model the precision of an item in IRT that, in contrast to traditional approaches, is considered to not be uniform across the range of values for which the knowledge is being measured. This function, based on Fisher information, quantifies the information of an item by applying the Eq. (3). In this equation, $P_i(\theta)$ is the probability distribution modeled by the ICC; $Q_i(\theta)$ is the opposite of the ICC, i.e., 1 - $P_i(\theta)$; and $\dot{P_i}(\theta)$ is the derivative of the ICC with respect to

the knowledge, θ. The information can also be quantified in tests by simply combining the different IIF of the items involved.

$$I_i(\theta) = \frac{\left[P_i'(\theta)\right]^2}{P_i(\theta)Q_i(\theta)} \tag{3}$$

IIF is the standard mechanism used in IRT to determine the reliability of the measurement instrument (Baker 2001). Since reliability is a statistic applied in a more general way to tests and this article is focused on items we avoid entering into further detail. Interested reader is referred to (Baker 2001; Hambleton et al. 1991) for the associated formulas and detailed explanations. The IIF is also used in adaptive testing in order to describe the quality of items and tests, which is used to apply item selection mechanisms and to compare different items and tests (Hambleton et al. 1991). Figure 2 shows three different shapes of the IIF.

A mechanism to quantify the information of a constraint is the area under the IIF curve (AUC) (Hambleton et al. 1991) that would be determined by the Eq. (4), where $A(i,\theta)$ represents the area of the IIF of the item i with respect to the knowledge.

$$A(i,\theta) = \int_{-\infty}^{\infty} I_i(\theta)\ d\theta \tag{4}$$

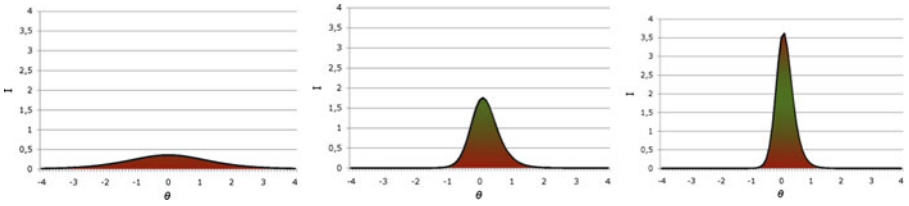
IRT is one of the most important test theories in psychometrics to measure psychological latent traits. In contrast to other theories, IRT focuses on characteristics of items, i.e., the ICCs. This allows the separation of the properties of items from the population where it is being applied. A brief summary of IRT and its advantages can be found in Meyer and Zhu (2013).

Evidence Centered Design

Evidence-Centered Design (ECD) is an approach for constructing educational assessments in terms of evidentiary arguments (Mislevy et al. 2004). The main goal of this proposal is to provide a framework to obtain inferences from what students say or do. The rationale of this approach is that tasks cannot be constructed in isolation, in the sense that we must provide mechanisms to assess them. The usefulness of a task will be related to how well it assesses the knowledge skills and abilities that students possess.

As can be seen in Figure 3, from a technical point of view, the ECD framework identifies three parts:

- The student model: This contains information on the student such as his/her knowledge state. It defines one or more variables related to what we want to measure.



Low IIF Medium IIF High IIF

Fig. 2 Different shapes of item information function

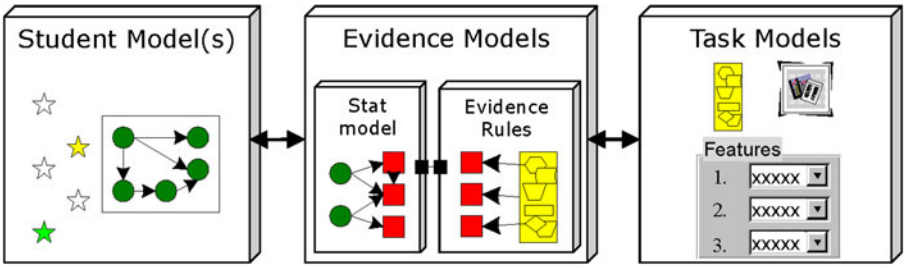


Fig. 3 Evidence-centered design framework (adapted from (Mislevy et al. 2004))

- The evidence model: This describes the evidence we will accept that the student possesses the targeted knowledge and how the students' performance in the tasks can be used to update their student model.
- The task model: This describes the material which is presented to the student and the steps he/she must carry out to generate a response.

Using ECD, students will interact with the task model performing one or more tasks. When a response is generated, this raw evidence will be used by the evidence model to infer the student's knowledge state, and update his/her student model accordingly.

This framework establishes a set of guidelines and generic elements without specifying a particular domain of application, assessment methodology, or restricting the form of the elements involved. ECD is a general framework. Developing a new system requires a different study with different structures, elements and methods. In short, it involves designing the student and domain models. Our research can be viewed as a special case of an ECD, but focuses on using a particular student modeling technique (the CBM) and a particular evidence model (the IRT). The task model should be implemented for each problem.

Related Work

Ensuring the quality of the parameters or rules used in the ITS has been a central point in the research area. In this section we have included some related proposals focused in two of the main student modelling techniques: the Cognitive Tutors (CT) and the CBM.

CT (Anderson et al. 1995) are based on a theory that claims that there are two long-term memory stores: declarative and procedural. Learning goes through several phases. The first involves learning declarative knowledge, including factual knowledge that later turns into procedural knowledge, which is goal-oriented and, therefore, more efficient to use. Procedural knowledge is represented in the form of production rules. In the last phase, the production rules are further optimised when the student becomes an expert.

Knowledge Tracing (KT) (Corbett and Anderson 1995) is a technique to model student knowledge and learning over time that have been primarily included in CT. It is used to predict the student performance based on the estimation of the probability of having learned the skills involved in the question resolution. It is based on the

estimation of four parameters: the initial knowledge, the learn rate and the guess and slip rate. Individual differences are achieved by defining a set of weights associated to each student that personalize the four parameters.

Learning Factor Analysis (Cen et al. 2006), propose a semi-automated method combining a multiple logistic regression model of learning with combinatory techniques to calculate the parameters of the model. Performance Factor Analysis (Pavlik et al. 2009) is a modification of the first to extend it with adaptation capabilities. IRT techniques has been previously applied to ITS. In the area of CT and KT there are some proposals of multidimensional IRT models (Cen et al. 2008). Pardos and Heffernan (2011) have extended the standard KT model to take into account different item difficulties. As they say: "*Models like IRT that take into account item difficulty are strong at prediction, and models such as KT that infer skills are useful for their cognitive diagnostic results*".

In CBM there are several approaches that try to study the elements in the domain model in order to improve the instructional quality (Martin and Mitrovic 2005, 2006; Martin et al. 2011). They all use learning curves with purposes very similar to the ones that will be presented in this study: a) detecting constraints that are not helping to improve learning; b) exploring different levels of generality in constraints that would produce better learning; and, c) comparing different systems based on their performance.

The fundamental part of these studies is to identify the elements of the learning curves not reflecting good learning. Once they have been detected, learning curves are checked for their generalizations or specializations and compared with the original ones. This allows the identification of different levels of generality that could lead to better instruction, which is used to present feedback. When changing the level of generality does not improve the learning curve, the constraint is considered to be a candidate for further review to explore the reasons why.

The authors of CBM studies state that learning curves can also be used to compare the performance of different tutors. To achieve this, a proportion of violated constraints should be averaged across all the subjects and all constraints. This allows an average learning curve to be generated that becomes the representative element to be used in a comparison with other systems.

The problem of the work discussed, is found in the use of heuristics to determine the probability of having learnt a constraint that is used in learning curves. Furthermore, they are designed to only model learning, they lack an assessment mechanism. This is covered in this research by the use of IRT and other statistical methods, which introduce well-founded methods to determine when a constraint presents an anomaly and to assess students. As far as we know, there is no application of IRT in the field of CBM.

Assessing the Quality of Constraints

A framework that combines IRT and CBM was initially proposed in (Gálvez et al. 2009a; 2009b). These papers represent the inception of the study being presented in this paper. This combined model extends the CBM paradigm with a long-term student model implemented using IRT assessment methods. Accordingly, the heuristics used to estimate student knowledge are replaced with a well-founded technique, which

results in a more accurate student model and, consequently, provides an improved adaptation to the student learning process. At the same time, this synergy opens a new door to IRT that allows assessments in problem solving learning environments.

The proposal was based on the analogy between *items* in IRT and *constraints* in CBM. Firstly, they are measurement instruments that, though in different environments, collect evidence of knowledge. Secondly, their nature and shape is the same because they are modeling declarative concepts of a particular domain. Next, both of them have two resulting values of the student performance: one positive and one negative. The positive value represents correct knowledge, which, in the case of CBM, corresponds to satisfaction of constraints and, in items, to a correct response. The negative value represents faulty knowledge, meaning that the constraint was violated or the response was wrong.

In addition to the equivalence between *items* and *constraints*, environments where they are applied, i.e., testing environments in IRT and problem solving learning environments in CBM, maintain a structural similarity that favours the use of the formal assessment methodology in the student modeling paradigm. This similarity is shown in Fig. 4. Here, it can be seen that a problem of a CBM tutor contains a set of constraints, which are related to relevant concepts to solve this problem correctly. The problem is equivalent to a test comprised of items, each one assessing the same concept as the corresponding constraint.

As with IRT, a Constraint Characteristic Curve (CCC) is defined for each constraint that represents a probability distribution based on the knowledge: the broader the knowledge, the more probability of satisfying the constraint. Violations can be modelled using the opposite of this curve, which means that when the knowledge is broader, the probability of violation is lower. The CCCs allow IRT mechanisms to be applied in CBM systems in the same way that they are applied in testing environments.

The power of CBM lies in the set of constraints that are incorporated inside the domain model. It is evident that a poor design of constraints has a negative effect on the instructional efficiency of CBM as well as other associated mechanisms. Therefore, a correct elicitation and careful design is necessary to ensure quality in the domain model, which is so important for producing accurate results. This is especially significant for the

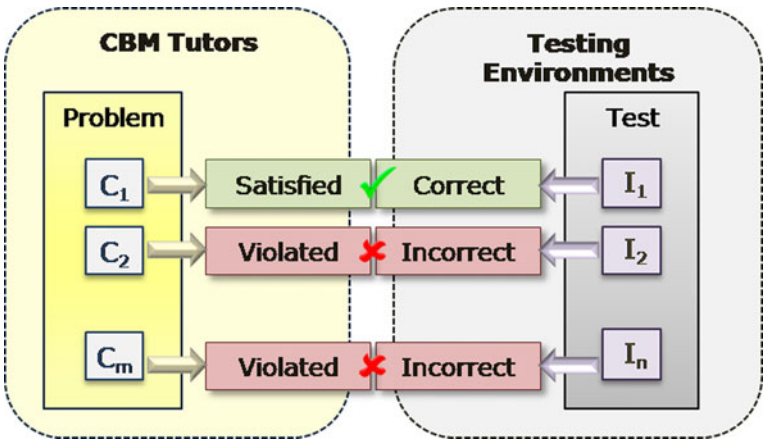


Fig. 4 Analogy between constraints and items

probabilistic model that combines CBM and IRT as estimation of knowledge could be extremely biased using a poor quality domain model. This section presents the sources of errors that can affect the assessment mechanism and the IRT method we propose to detect them and guide the assessment towards more accurate estimates.

Sources of Error in Constraints Used for Assessment

There are two phases in the CBM + IRT probabilistic model where errors could appear and, thus, it results in constraints that are inappropriate for assessment. The first phase is found in the very first step conducted to apply CBM: the encoding process of constraints as inference rules. If a constraint is not correctly encoded, it becomes a measurement instrument that produces incorrect knowledge estimates. This situation was experimentally discovered while testing the validity of the mechanism to detect inappropriate constraints produced during the second phase mentioned below. With respect to this phase, there are two possible sources of errors:

- The first possible error occurs when a constraint has been encoded incorrectly due to a mistake made by the expert. A negligible failure in the encoding process can make the inference rule behave in a completely different way to the original intended behaviour, resulting in a constraint with no quality.
- Even if the principle is encoded correctly, there may still be other errors associated with the level of generality of the domain principles: a constraint can be modelled according to a more general principle or a more specific one. Normally, the most suitable level of generality cannot be easily determined using objective criteria and, therefore, there might be ambiguity or controversy. Consequently, the decision on the specificity of the constraint has a more subjective component and it is influenced by the constraint designer's criterion.

Regarding the latter error, as mentioned in the related work section, authors of CBM tutors have already conducted studies on the granularity of concepts associated with constraints. They concluded in these studies that groups of constraints, linked to more general concepts, would be more effective for learning than single constraints in some cases and, in others, splitting constraints into more specific concepts was more effective (Martin and Mitrovic 2005, 2006; Martin et al. 2011). This reflects that generality is also a factor to be taken into account, to provide a domain model with quality both, in the original application of CBM for instruction and also in the assessment conducted using IRT.

The second phase where errors can occur is associated with the application of assessment mechanisms to determine the knowledge. The reason is that it is not related to the validity of IRT mechanisms, but with the amount of evidence available. In this case, the source of the error itself is the shortage of evidence and affects the two main stages of the assessment process:

- In the first stage, where the CCCs are calibrated, if the calibration algorithm receives no evidence as input, it is not possible to say anything about the constraint validity.
- In the second stage, the assessment process combines the probability distribution of the CCCs using a multiplication operation. The result of multiplying using an inappropriate curve biases the resulting distribution and, thus, produces poor

knowledge estimation. If a constraint is always violated, or if it is never violated by any student, then the information provided by that constraint is not relevant for the student assessment, and it only introduces noise.

Proposed Mechanisms to Study Quality of Constraints

The basis of our proposal to detect the two sources of error discussed in the previous section is that, considering the parameters of a CCC, we can manage constraints as if they were items and, consequently, mechanisms applied over items to determine their quality are equally valid for constraints. Psychometrics and psychological measurement theories like IRT use different techniques to analyse item quality: (a) Point-Biserial Correlation; (b) Analysis of influence in Cronbach's alpha; and (c) Analysis of the Item Information Functions.

Point-Biserial Correlation (PBC) (Lev 1949) is a classical way of analysing dichotomous variables. In our case, it can be defined as the correlation between the performance outcome (satisfaction or violation) of a given constraint and the final problem score θ_j. PBC can be a primary source of information to validate a constraint. The PBC should be positive for all constraints. A negative value would suggest that students with higher scores violate the constraint and, accordingly, it suggests that the constraint was designed incorrectly. High positive values indicate a correct behaviour. The PBC of a constraint i can be defined as:

$$r_i = \frac{\mu_1 - \mu_0}{\sigma}\sqrt{\frac{n_1 n_0}{n(n-1)}} \tag{5}$$

where σ is the standard deviation of the total scores θ_j; μ_1 and μ_0 are the mean of the scores θ_j when the constraint was satisfied or violated respectively; n, the whole sample size; and$n_1$and n_0,the number of violated and satisfied constraints, respectively.

Cronbach's Alpha (Cronbach 1951) is a measure of the internal consistency of a set of items. In our proposal, it can be defined over the sum of all constraints C_1, C_2,... C_n, in a problem by the formula:

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k}\sigma_{c_i}^2}{\sigma^2}\right) \tag{6}$$

where σ is the standard deviation of the total scores, and σ_{C_i} the standard deviation of the constraint i for the current sample of students. It is a common practice to use the percentage of correct responses, (in our case, the percentage of satisfied constraints), as an estimator of the total score. For this reason, the Cronbach's alpha does not require a previous calibration of the CCC parameters.

Cronbach's alpha is directly related to the average of the PBC of the constraint involved in the problem. It can be used to detect the influence of a given constraint following an iterative procedure that consists in removing every constraint and recalculating the alpha coefficient. After removing a constraint, if the alpha increases, it means that the problem consistency increases, and could suggest that the constraint

is not aligned with the others. It might be an incorrect constraint or a constraint measuring a different concept (too general, or too specific). The process would be repeated with the remaining constraints, selecting one by one a candidate to be removed. The process might end under three circumstances: (a) After removing a constraint, the Cronbach's alpha decreases or the increment is not significant. (b) Certain Cronbach's alpha threshold is reached (for instance, a value higher than 0.7 or 0.8 is commonly accepted as a good reliability). (c) The number of remaining constraints has reached a low threshold.

We can also define the Constraint Information Function (CIF) that can be used to detect the most suitable constraints for assessment. In this way, assessment can be carried out over concepts that more faithfully represent the reality, which reduces misleading results from an inappropriate representation. The expression of the CIF is obtained for constraints using CCCs instead of ICCs, which requires performing the calibration process beforehand. Since in this work we have focused on using the 3PL to model the CCCs, the CIF is calculated using Eq. (1) within Eq. (3), which after being reduced, according to Birnbaum (1968), produces the expression of the following equation:

$$I_i(\theta) = \frac{2.89a_i^2(1-c_i)}{[c_i + e^{1.7a_i(\theta-b_i)}][1 + e^{-1.7a_i(\theta-b_i)}]^2} \tag{7}$$

Here, $I_i(\theta)$ represents how informative a constraint i is, depending on the value of the student's knowledge θ. This knowledge ranges from -∞ to ∞ but, in practice, normally only values from the interval [−4, 4] or [−3, 3] are considered because, beyond this interval, the value of the CIF is closer 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 located in $\theta=b_i$, which is the parameter corresponding to the difficulty of the constraint and the most representative for the CIF. If the guessing parameter (c_i) is very high the maximum is slightly moved but is still close to the difficulty value. The information obtained after applying the CIF can vary according to the calibration of the CCCs.

The shape of the CIF is mainly explained by the discrimination factor (a_i) but the guessing factor (c_i) plays an important role when it is high. The difficulty parameter (b_i) only has an influence the relative position in the θ axis. The Problem Information Function (PIF) can be defined as the sum of all CIF involved in the problem. A well-distributed set of values of the difficulty parameter is important in order to ensure a good assessment for different knowledge levels.

In order to determine whether a curve is high or low, a comparison with the CIF of other constraints is required. Unless all constraints have an anomaly or little evidence, abnormal situations will be detected. By examining the AUC value obtained after applying Eq. (4) to Eq. (7), we can detect when a constraint is of poor quality due to an error. Each of the two possible extreme values is indicative of different error situations:

1. Extremely high values for a constraint, in comparison to the others, could suggest that this constraint is grouping more than one domain principle. This error is produced during the elicitation phase due to an incorrect generality level. The recommendation here, in order to get a better domain model, is to split this

constraint into several ones, each one modelling a more specific principle. An example of this situation is presented in Fig. 5. The labels CIF-1 and CIF-2 correspond to two different constraints. If these constraints are combined, the resulting CIF has a higher shape. This is due to the fact that a combined constraint can only be satisfied or violated (a dichotomous value) while two separated constraints can have four values; consequently if they are applied separately they provide more information. Notice that the total information obtained applying two constraints are greater than the information obtained with a single combined constraint, unless the constraints were wrong or misleading. The information function of the single combined constraint is the product, while the total information function of two constraints is the sum of both, which is (in general) higher. However, the decision to split or not a constraint will depend on the semantic of that constraint and the concepts involved.

2. Extremely low values could suggest that the associated constraint is not providing much information. In this case the source of the error could have been produced by two situations:

   a. There is not enough evidence to produce a good calibration for that constraint. This fact could be due to a small population or because constraints are not relevant in the set of problems. In either of these cases, the constraints are not providing enough evidence and they should be discarded during the assessment process.
   b. The constraint is too fine-grained due to an incorrect level of generality introduced during the elicitation phase. In this case, the constraint could be merged with another constraint (or constraints), in order to model a more

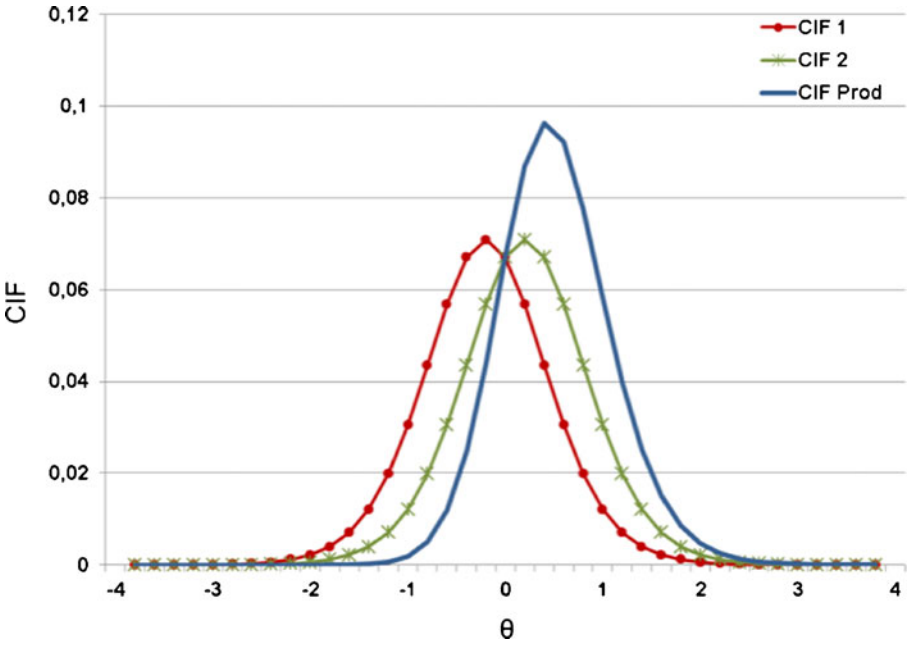


Fig. 5 An example of a composed constraint to be split

general principle. Figure 6 exemplifies this situation. Constraints C_1 and C_2 has a low CIF and, therefore, by combining them a better constraint is obtained. However, the combination would only be possible according to the constraints meaning and the associated concepts. It is clear that any two pair of constraints cannot be merged if they are associated to different and unrelated concepts.

Regarding the distinction between good and bad constraints, it is clear that the lower the amount of information, the worse they will be for assessment. Nevertheless, if we would have to set a threshold to distinguish between good and bad constraints, its value should probably depend on the domain. Nevertheless, the analysis based on the information function just points out the focus of attention, a semantic understanding of the constraints is needed in order to decide if a constraint could be kept, discarded, split or if it might be combined with another.

Tools Used in the Experiment

To perform the experiment we used three systems, each one for a different purpose: the first one is Siette (Conejo et al. 2004; Guzmán et al. 2007), 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 2010). Following the philosophy of this framework, each component is independent and can communicate through Web Services with the rest of the platform's components. These components are called Project Investment Problem Solving Environment (PIPSE) and CBM-Engine.

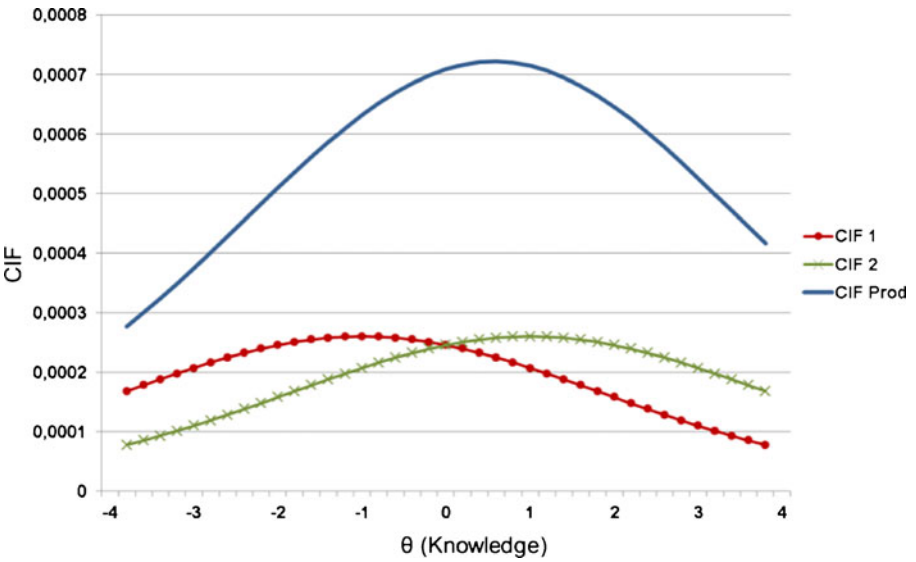


Fig. 6 An example of two constraints and the result after combined

Project Investment Problem Solving Environment

PIPSE was developed to be used as part of a course in Project Management as a support tool. It is a problem solving environment that focuses on the study of the profitability of starting up a project given a series of variables associated with the costs and benefits that it could generate. The system is a Web application implemented on Microsoft.NET through which students can apply several indexes, such as Net Present Value (NPV) or Internal Rate of Return (IRR) (Khan and Jain 1999), to study the profitability of a project. Figure 7 shows the four main parts of PIPSE: A is a panel of actions related to the current session and to the student's attempts to solve the problem; 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.

The system interface tries to reduce the cognitive overload (Sweller et al. 1998), otherwise calculus inherent in these kinds 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 outside the interface unnecessary 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, taken as a whole, represents the solution to the problem. PIPSE is able to present information about the

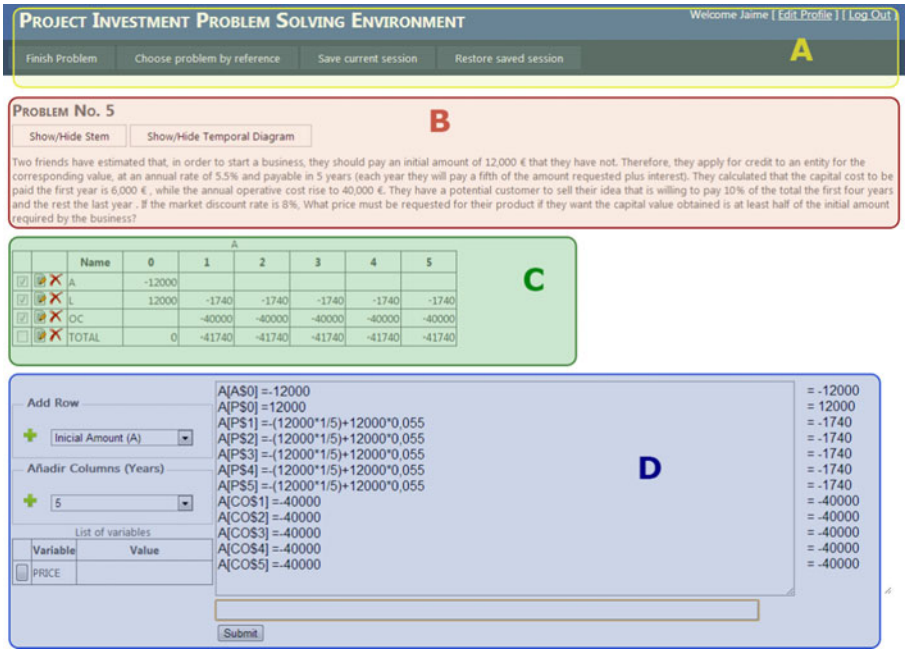


Fig. 7 Project investment problem solving environment

student's 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's 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 sub-systems 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.

CBM-Engine Assessment Component

The CBM-Engine is a Service Oriented Architecture-based component following the same idea as (Gálvez et al. 2008) that implements CBM with IRT assessment. The core of this component is a set of services that can be used to apply the previously explained methodology in any external system/tutor. New problem solving environments or tutors wanting to use the assessment provided by this framework must be registered using an authoring tool. The authoring tool, which is still being improved, allows the definition of constraints and data structures that will provide the information associated with the solutions.

The architecture of a CBM-Engine, depicted in Fig. 8, is formed by a three-layered architecture comprising: a) a top level services layer offering Web Services as an interface with the external systems, b) a business logic 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. Although the framework has a Web interface, its sole purpose is to perform basic administrative tasks and, therefore, it is not included as a relevant part of the architecture.

An external system such as PIPSE only communicates with the top Layer using pieces of information that we have called *external structures* (in Fig. 8, they are labelled with *ES* and highlighted with a dashed outline). This information can only be sent to CBM-Engine via three SOA entry points, which groups services according to their functionality:

- First, the management entry point, allows an external system to manage users and problems. The utilization of these services is fundamental in order maintain the problems set and to control user access.
- The Session entry point is in charge of gathering evidence of external systems during a session. Through services belonging to this entry point an external system manages the session, submits information related to the solution, and invokes CBM process to check violations of constraints.
- The IRT entry point contains the services associated with the IRT mechanisms already explained.

Every entry point has associated with it, an internal module in the business logic layer that is in charge of handling the information that comes from PIPSE and producing the corresponding response, which could be the list of violated constraints

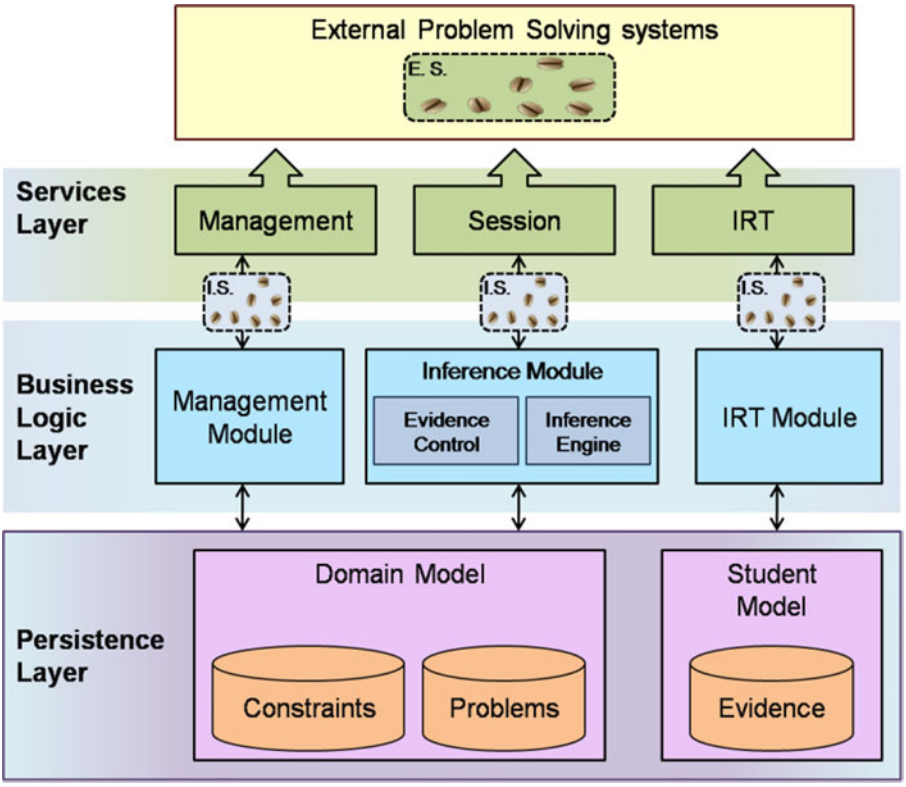


Fig. 8 CBM-engine architecture

or the result of IRT mechanisms. In order to accomplish their function, internal modules interact between them and use the elements stored in the lower layer of the CBM-Engine. These elements are the problems and constraints of the domain model, and the content of the student model.

In the particular case of the PIPSE system, according to the categorization made by Mitrovic and Weerasinghe (2009), we are dealing with a well-defined domain where problems as well as tasks are well-defined. Therefore, the size of the domain model is small comprising a set of 17 constraints that are added to the engine with the required specific data structures. These constrains can be categorized into three subsets according to different conceptualizations: (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. An example of a PIPSE constraint belonging to category c) is represented in Fig. 9. Its original encoding has been adapted to natural language to

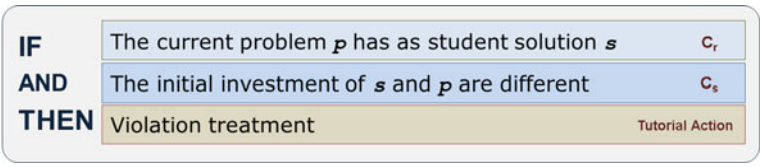


Fig. 9 PIPSE sample constraint

make understandable to the reader. The associated concept in this constraint is the initial investment done in a project p. Detection of mistakes is done through the relevance and satisfaction conditions (C_r and C_s respectively) by comparing the current student solution with the teacher solution. If that is the case, the violation treatment is fired.

Experimentation

In this section we are going to describe two experiments we have conducted to validate our proposal. In this respect, the main hypothesis to be tested will be whether or not the three proposed methods, i.e., PCB, Cronbach's alpha and CIF analysis can be applied to constraints in the same way they are used in testing environments, to detect constraints unsuitable for assessment.

Initially, the whole set of 17 constraints was used in each experiment to check the correctness of the solutions. However, the problems proposed in experiment 1 and 2 were different and, thus, the subset of constraints involved in each experiment. This subset is formed by those constraints that are relevant in the problems presented in an experiment. Some of the constraints were only relevant in one of the experiments while others appeared in both of them.

Design and Implementation of the Experiments

In order to evaluate the efficiency of our methodology, we conducted a **first experiment** with undergraduate students in the last year of a M.Sc. in Computer Science at the University of Malaga. A total of 24 students participated in the study that was carried out in December 2011 and comprised several stages. First, the students were instructed over several classes on the different indexes to solve the project investment problems. Next, they undertook a one-hour-long session where they were able to use the system to solve two problems that had been seen previously in class; a week later, we set them a paper-based exam where two problems were posed.

To explore the experiment hypothesis, problems posed in the exam did not cover the whole set of constraints. Unlike our early work with this technique, the exam was set on paper with the aim to get only the constraint violations and to prevent 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 the constraints were checked against them.

Theoretically, in an environment without learning happening such as the paper exam used in our experimentation, if the student knows the concepts associated with a constraint, no violation should occur in the whole session. For this reason, the outcomes or instances of a constraint can be considered the same item across different problems of the experiment. In this way, if a constraint was violated in any of the two problems, then it is considered to be violated by the student in the experiment.

Presentation of the exam in a paper environment also simplifies the experimentation since only one attempt can be submitted for each problem. This means that a

constraint would appear between 0 and 2 times in an experiment, depending on its relevance in each of the two problems.

The experiment was used as an exam within the course, and all 24 students enrolled on the course participated. Once all data were gathered from students, we performed the analysis of constraints applying the approaches described before and later studying their validity.

A **second experiment** was also conducted to further explore the preliminary results obtained in the first one. Its design was quite similar to the former. Consequently, this new experiment consists of the same stages, has the same initial set of constraints, and the motivation was also the same. Nevertheless, it had some slight differences regarding the previous experiment. Although it was conducted for the same subject matter, it was performed on another academic year (in January 2013) with a different student sample. This time a total of 23 students participated in the study. Secondly, the two problems to be solved were chosen by the course lecturers in order to cover the maximum number of different constraints as possible regarding the first experiment. The rationale of this was to explore whether or not our methods are consistent over different datasets.

Results

This part reviews and discusses results according to the main goal of the experimentation. The results obtained for the two experiments are presented in separate sections.

First Experiment

In this experiment the initial set of 17 constraints was reduced to 7. The rest did not appear in the problem or were answered uniformly by all students (either satisfied or violated). Accordingly, only 7 constraints provide enough information to be analysed.

In this experiment, the PBC technique produced the result in Table 1, where we can see that C_2 has a significant lower value than the others. C_6 also has a low value and would be a candidate to be discarded from the dataset.

The Cronbach's alpha obtained for the dataset of this experiment was 0.557 considering the 7 constraints. For every constraint we examined whether or not removing it from the dataset would produce a more consistence result. The Cronbach's alpha obtained when a given constraint is removed is summarized in Table 2. The results indicate that Cronbach's alpha will increase to 0.622 if C_2 is removed.

After removing this constraint, we explored which constraint would be negatively affecting the index following the same process as before: the new Cronbach's alpha is calculated after removing every remaining constraint, resulting in the Table 3. It can be seen that constraint C_6 could be removed and consistency would

Table 1 Point Biserial Correlation for the data from experiment 1

Constraint	C_1	C_2	C_6	C_8	C_9	C_{14}	C_{17}
PBC	0.542	**0.210**	**0.364**	0.695	0.411	0.674	0.542

Table 2 Value of Cronbach's alpha that is obtained removing a single constraint in experiment 1. First iteration

Constraint	C_1	C_2	C_6	C_8	C_9	C_{14}	C_{17}
Cronbach's alpha removing C_i	0.503	**0.622**	0.568	0.446	0.517	0.441	0.503

increase. Nevertheless the improvement is not significant and, therefore, we stopped in this iteration.

The solution provided by each student was introduced into PIPSE, which sent it to CBM-Engine, recording all data and calibrating constraints separately for each experiment. The calibration output, i.e., parameters representing the CCC, was analysed by applying the information function to each constraint using the formula of Eq. (4), which produced the values of Table 4. As a result, in this experiment we obtained an average value of 14.808 of the AUC and a standard deviation of 2.186 for the whole set of constraints.

Constraint C_2, which was detected by PBC and Cronbach's alpha techniques as a questionable constraint, was also the one with a lower value of AUC, followed by constraint C_6. Those constraints were also those that have a lower value of the discriminant factor (a) and a slightly higher value of the guessing factor (c).

We can also formulate a manual method that tries to discard constraints visually using the graphic representation of the CIF. This method suggests removing those constraints providing information already supplied by other constraints. The key of this method is to select those constraints able to cover as much knowledge levels as possible, avoiding redundant information. For example, looking at the constraints in Fig. 10 we can see that, although there are some gaps in the whole range of knowledge, CIF curves corresponding to C_2 and C_6 are under other curves. They are providing information already supplied by C_1, C_{17} and C_{14}. Those constraints are clearly not good to get a non-redundant dataset and it is also coherent with the results obtained from the methods studied previously. The remaining five constraints C_1, C_8, C_9, C_{14} and C_{17} are performing well.

If we represent the PIF grouping the whole set of constraints, we can examine the effect of each constraint in this function. The Fig. 11 shows that removing constraints C_2 and C_6 only reduce the kurtosis of the function, whereas removing constraints covering knowledge ranges not covered by others, would lead to changes in the function shape and, thus, to a loss of information. That is precisely the case of C_8: removing it, will affect seriously to the knowledge range [1, 2].

Constraint C_8 exhibits the highest difficulty and CIF (see Table 4 and Fig. 10). This might suggest that this constraint could be a good candidate to be divided. The same can be said for constraint C_{14}.

Table 3 Value of Cronbach's alpha that is obtained removing a single constraint in experiment 1. Second iteration

Constraint	C_1	C_6	C_8	C_9	C_{14}	C_{17}
Cronbach's alpha removing C_i	0.564	**0.634**	0.568	0.586	0.541	0.564

Table 4 List of constraints parameters in the first experiment

Constraint Name	ID	a	b	c	AUC
A value is wrong	C_1	1.89	−0.58	0.25	17.283
Number years wrong	C_2	1.3	−0.75	0.28	11.159
Sell value is wrong	C_6	1.56	−0.66	0.27	13.677
General solution wrong	C_8	1.7	1.64	0.13	20.065
Operative cost value is wrong	C_9	1.79	−1.06	0.26	16.027
Salary value is wrong	C_{14}	1.95	−0.01	0.23	18.597
Operative costs doesn't exist	C_{17}	1.89	−0.58	0.25	17.283

Second Experiment

In this experiment, the initial set of 17 constraints was reduced to 6. The rest did not appear in the problem or were answered uniformly by all students (either satisfied or violated). As a consequence, only 6 constraints provided enough information to be analysed.

Using the same analysis as the previous experiment, the PBC obtained is shown in Table 5. The results do indicate a problem with C_{12}, since its PBC is negative. Other constraint under suspicion is C_9due to its low value in comparison to the others. Nothing can be said of the rest of the constraints.

The initial Cronbach's alpha obtained for this dataset was 0.219. The increment of the index after removing every constraint is represented in Table 6. According to these results, it is clear that removing C_{12} would improve considerable the consistence to 0.327.Moreover, we can see a slight improvement with constraint C_9 and almost negligible with C_2.

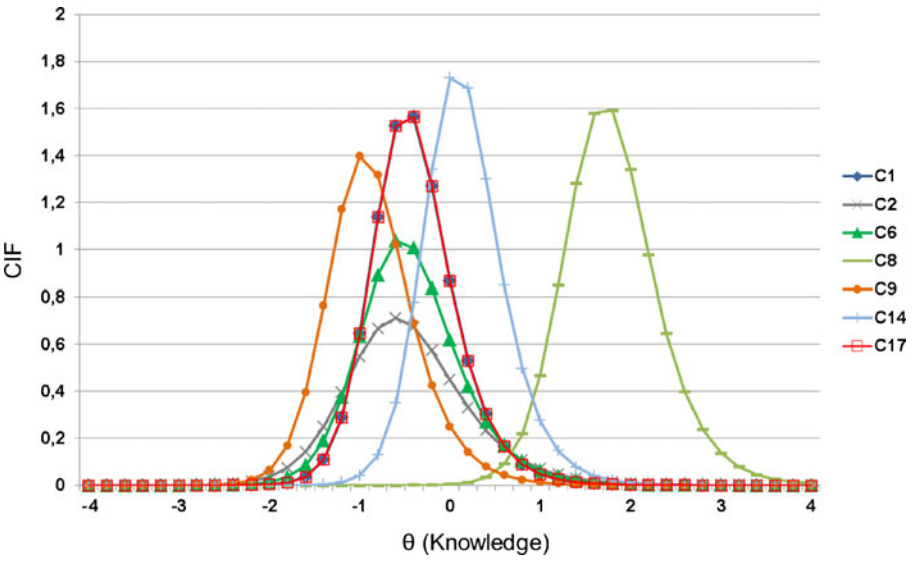


Fig. 10 CIFs of every relevant constraint in experiment 1

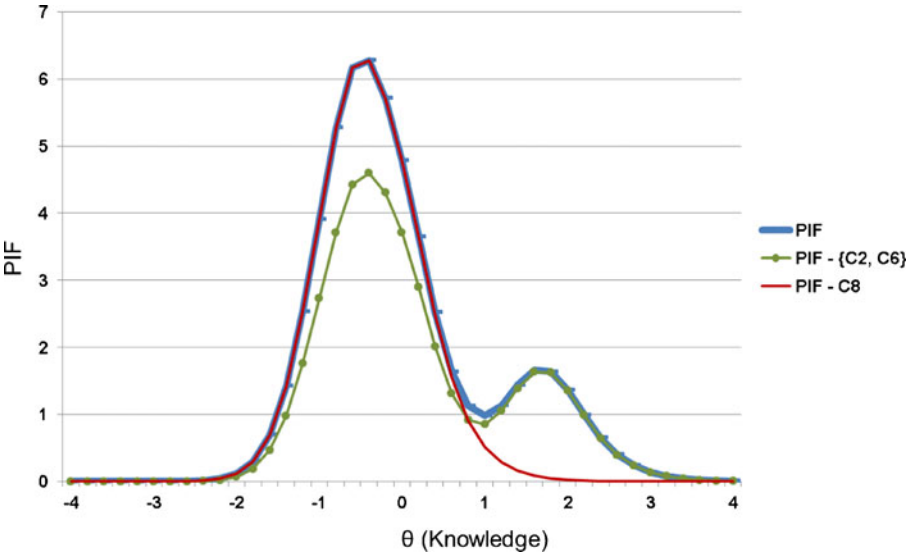


Fig. 11 Variants of PIF in experiment 1 after removing certain constraints

We continued the analysis by removing this constraint and the process was repeated producing the values of Table 7. In this case, the previous remarked constraint, C_9, was detected as the next candidate to be removed. We decided to stop here because in this iteration the improvement was insignificant and the number of remaining constraints is reduced. In this latter case, keeping removing constraints could produce over fitting.

In order to compare the previous methods with the result obtained using the CIF, the Table 8 shows the AUC values of the constraints in the second experiment. Using this information we can see that the set of candidate constraints to be removed, according to PBC and Cronbach's alpha (C_{12} and maybe C_9), matches with the set suggested by the AUC analysis.

Figure 12 shows the CIF for all those constraints providing some evidence. We can see clearly that there is a strange constraint and other that is redundant. C_{12} is much more flattened than the other constraints and its shape is quite strange. At the same time, C_9 is focused in the range [−1, 1], which is precisely covered by C_{17}. These two constraints would be candidates to be removed, which is also consistent with the result of previous methods.

Looking at the PIF of 2013 dataset, depicted in Fig. 13, we can also see the effect of removing a good constraint. If we remove C_9 and C_{12}, the shape of the curve is the same but it is slightly flattened. Nevertheless, removing C_1, which is good, affects loosing information from the range [−0.5, 1.5].

Table 5 PBC for the data from experiment 2

Constraint	C_1	C_2	C_8	C_9	C_{12}	C_{17}
PBC.	0.647	0.444	0.334	**0.282**	**−0.054**	0.709

Table 6 Value of Cronbach's alpha that is obtained removing a single constraint in experiment 2. First iteration

Constraint	C_1	C_2	C_8	C_9	C_{12}	C_{17}
Cronbach's alpha removing C_i	0	0.22	0.124	0.249	**0.327**	0.093

Finally, a comparison of the results of experiments 1 and 2 is presented in Table 9. The main finding is that the three mechanisms give similar results taking into account the experiments in which they are applied. Some constraints were never triggered neither in experiment 1 nor 2, or they were always producing the same result (violation or satisfaction) for all students. Nothing can be said about these constraints, but the lack of data suggests that they should be revised. Constraints C_1, C_2, C_8, C_9 and C_{17} were shared in both experiments. Three of them (C_1, C_8 and C_{17}) have been diagnosed as good constraint in both experiments. The other two, are also valid.

A bad constraint, C_{12}, has been detected in experiment 2. A more detailed inspection of that constraint revealed the possible cause. That constraint checks whether or not the student introduced an initial value of the investment. The problem might have been solved without introducing that value. Thus, violating this constraint might be explained by a slip of the student, and not by a low knowledge.

Discussion

At a first glance, the experiments show that only 8 out of the initial set of 17 constraints played an active role in the problem-based assessment. The other 9 were never triggered, or they were always satisfied (or violated) by the students. This was an unexpected result that requires: (a) planning further experiments with other problems; (b) to review the interface and/or the constraints to check that they are triggered when they are relevant.

Results obtained in the first experiment suggest that good and bad quality constraints could indeed be detected using statistical IRT methods. In this experiment two constraints were detected as anomalous. We have also detected a constraint with a high difficulty and CIF. This fact revealed that the constraint grouped several concepts and should be split into others two representing more fine-grained principles of the domain.

In the second experiment, the PBC method detected an incorrect constraint. Its inspection revealed that a student slip might have been the cause of violating that

Table 7 Value of Cronbach's alpha that is obtained removing a single constraint in experiment 2. Second iteration

Constraint	C_1	C_2	C_8	C_9	C_{17}
Cronbach's alpha removing C_i	0.101	0.343	0.265	**0.373**	0.228

Table 8 List of constraint parameters in the second experiment

Constraint Name	ID	a	b	c	AUC
A value is wrong	C_1	1.1	0.39	0.24	10.257
Number years wrong	C_2	0.92	−2.04	0.27	8.032
General solution wrong	C_8	0.81	2.83	0.08	8.77
Operative cost value is wrong	C_9	0.63	−1.14	0.29	5.243
A doesn't exist	C_{12}	0.47	−3.35	0.27	3.222
Operative costs doesn't exist	C_{17}	1.11	−0.93	0.25	10.149

constraint, even though the student knowledge was high. This fact suggests improving the interface to avoid slip.

The results found in the second experiment are consistent with those obtained for the first one. The three techniques proposed to diagnose the quality of constraint perform well and gave consistent result. These techniques complement each other. PBC has revealed to be very useful in order to detect faulty constraints. Cronbach's alpha highlights the constraints that are more inconsistent with the other, requesting attention on those. Lastly, the analysis of the information function produces an overall view of the behaviour of each constraint with respect to the others and to the whole problem. Decision of what constraint should be kept and what one should be removed or modified must be taken into account considering the whole view.

Larger datasets are needed to achieve definitive conclusion about the constraints. The development of a good constraint set is a continuous task. The techniques proposed in this paper provide statistical evidence, but are limited to the data available. The statistical analysis proposed in this paper should be considered just as a recommendation of what constraint should be checked. The

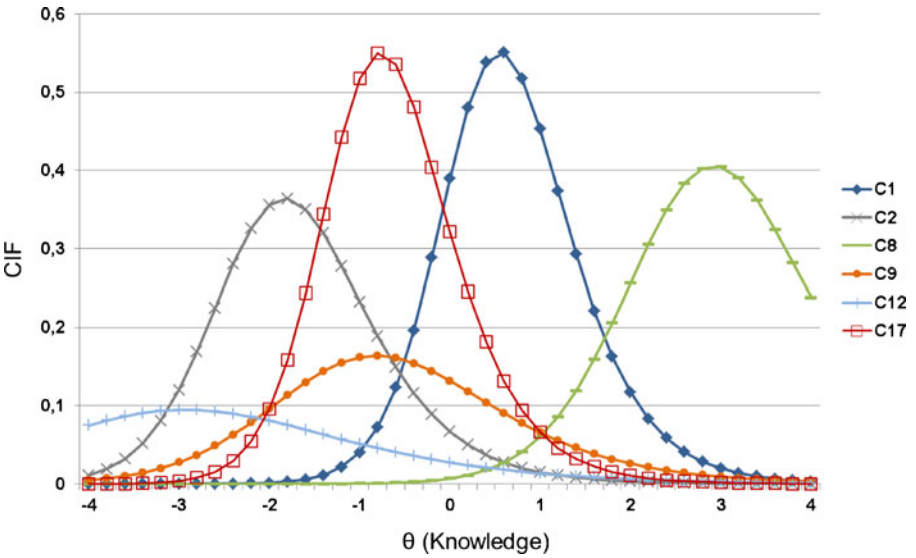


Fig. 12 Variants of PIF in experiment 1 after removing certain constraints

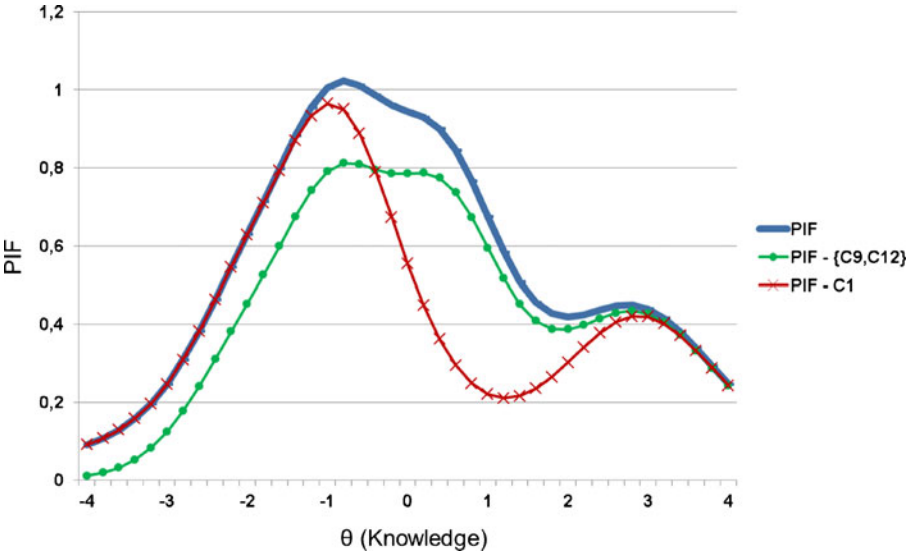


Fig. 13 Variants of PIF in experiment 2 after removing certain constraints

method will be especially useful for large set of constraints, where the identification of suspects will save development time.

Table 9 A comparison of results of experiments 1 and 2

		Experiment 1			Experiment 2		
Constraint Name	ID	PBC	CA	CIF	PBC	CA	CIF
A value is wrong	C_1	High	Keep	V.Good	High	Keep	V.Good
Number years wrong	C_2	Low	Reject	Low	Medium	Keep	Medium
General range solution wrong	C_3	NA	NA	NA	NA	NA	NA
Investment value is wrong	C_4	NA	NA	NA	NA	NA	NA
Loan value is wrong	C_5	NA	NA	NA	NA	NA	NA
Sell value is wrong	C_6	Medium	Keep	Bad	NA	NA	NA
Problem doesn't have general solution	C_7	NA	NA	NA	NA	NA	NA
General solution wrong	C_8	High	Keep	V.Good	Medium	Keep	Good
Operative cost value is wrong	C_9	Medium	Keep	Medium	Medium	Keep	Medium
Salary doesn't exist	C_{10}	NA	NA	NA	NA	NA	NA
Sell doesn't exist	C_{11}	NA	NA	NA	NA	NA	NA
A doesn't exist	C_{12}	NA	NA	NA	V. low	Reject	Bad
Investment doesn't exist	C_{13}	NA	NA	NA	NA	NA	NA
Salary value is wrong	C_{14}	High	Keep	Good	NA	NA	NA
Problem doesn't have general range solution	C_{15}	NA	NA	NA	NA	NA	NA
Loan doesn't exist	C_{16}	NA	NA	NA	NA	NA	NA
Operative costs doesn't exist	C_{17}	High	Keep	Good	High	Keep	Good

Conclusions

This paper explores how the quality of constraints in CBM tutors can be analysed using three different techniques commonly used in IRT. This methodology is based on the analogy between test items and constraints. This approach could help generate better tutors and more accurate assessment, leading to a more precise student model and a better adaptation.

The three approaches include the analysis of the PBC, the Cronbach's alpha and the CIF. The performance of the three methods have been analysed in two experiments with real data obtained from the PIPSE problem-solving environment in the domain of project investment analysis. Results suggest that the three methods are able to detect faulty, bad and good quality constraints. The analysis of CIF seems to be more general that the two others, which have, however, the advantage of being easier to apply. In addition, the CIF method could contribute to the constraint elicitation process, since it could help detect constraints that should be split or grouped, and provide information to decide whether to reformulate or discard them. Students' data were analysed to produce first a calibration of constraints and then an assessment.

Nonetheless, each of the three proposed methods has limitations. PBC performs very well detecting faulty constraints, but does not provide much information about good quality constraints. Analysis of Cronbach's alpha does not guarantee that the removed constraints were incorrect. Moreover, if constraints are removed to artificially get an alpha too high, then it may suggests a high level of constraints redundancy; that is, the remaining constraints are asking the same question in slightly different ways. The CIF might be more difficult to calculate, because it requires the calibration of the constraints, but it provides a better understanding of the role of each constraint in the problem. However, in our experiments the three methods are consistent and give similar results.

A third failed experiment, not included in this paper was designed in order to compare assessment obtained with the problem solving environment and the assessment resulting from a test using with multiple choice questions. The two types of assessment were not statistically significant but there was a low correlation between them. Those results might be explained by the two assessments measuring a different type of skill since knowledge required to answer a question is more theoretical than and not as practical as that to solve a problem. Due to this issue, a carefully designed new set of experiments is required.

To be used in this 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.

The results are promising and a range of possibilities open up with this synergy. The results of the pilot studies included in this paper encourage us to explore the effectiveness of this technique in bigger systems. Further work should also be done to explore whether the process of the approach presented here, which was entirely manual, could be automated within the CBM-Engine or within Siette assessment framework that already implements some of the proposed IRT item analysis. Our plans also include testing whether or not some other properties of the information function, such as the maximum value or the kurtosis, could lead to a better diagnosis of the quality of the constraints.

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

References

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: lessons learned. *The Journal of the Learning Sciences, 4*(2), 167–207.

Baker, F. (2001). The basics of item response theory. In *ERIC Clearinghouse on assessment and evaluation*. College Park: University of Maryland.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In *Statistical theories of mental test scores*. Reading: Addison-Wesley.

Cen, H., Koedinger, K. R. & Junker, B. (2006). Learning factors analysis - a general method for cognitive model evaluation and improvement. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, pp. 164–175.

Cen, H., Koedinger, K.R., Junker, B., (2008) Comparing two IRT models for conjunctive skills, In: Woolf, B., Aimer, E., Nkambou, R. (eds.): *Proceedings of the Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, Montreal, Canada

Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-de-la-Cruz, J. L., & Ríos, A. (2004). Siette: a web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education, 14*(1), 29–61.

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*, 253–278.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika, 16*(3), 297–334.

DEDALO project (2010), Assessment and Learning of Mathematics. http://dedalo.lcc.uma.es (Accessed on September 14, 2013).

DeMars, C. (2010). Item response theory. In *Series in understanding statistics: Measurement*. USA: Oxford University Press.

Gálvez, J., Guzmán, E., & Conejo, R. (2008). A SOA-Based Framework for Constructing Problem Solving Environments. *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies*, pp. 126–127.

Gálvez, J., Guzmán, E., & Conejo, R. (2009a). Data-driven student knowledge assessment through Ill-defined procedural tasks. Current Topics in Artificial Intelligence, CAEPIA-2009 Selected Papers, Lecture Notes in Artificial Intelligence, 5988, 233–241. CAEPIA-TTIA 2009.

Gálvez, J., Guzmán, E., Conejo, R., & Millán, E. (2009b). Student knowledge diagnosis using item response theory and constraint-based modeling. *The 14th International Conference on Artificial Intelligence in Education, 200*, 291–298.

Gálvez, J., Guzmán, E., & Conejo, R. (2012). Exploring Quality of Constraints for Assessment in Problem Solving Environments. *The 11th International Conference on Intelligent Tutoring Systems*, pp. 310–319.

Guzmán, E., Conejo, R., & Pérez-de-la-Cruz, J. L. (2007). Improving student performance using self-assessment tests. *IEEE Intelligent Systems, 22*, 46–52.

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of item response theory.* Thousand Oaks: Sage Publications, Inc.

Khan, M. & Jain, P. (1999). *Theory and Problems in Financial Management*. McGraw Hill Education.

Lev, J. (1949). The point biserial coefficient of correlation. *Annals of Mathematical Statistics, 20*, 125–126.

Martin, B. & Mitrovic, A. (2005). Using learning curves to mine student models. *10th International Conference on User Modeling*, pp. 79–88.

Martin, B., & Mitrovic, A. (2006). The effect of adapting feedback generality in ITS. *Adaptive Hypermedia and Adaptive Web-Based Systems, 4018*, 192–202.

Martin, B., Mitrovic, A., Koedinger, K. R., & Mathan, S. (2011). Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction, 21*(3), 249–283.

Mayo, M., & Mitrovic, A. (2001). Optimising its behaviour with bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education, 12*, 124–153.

Meyer, J. P., & Zhu, S. (2013). Fair and equitable measurement of student learning in MOOCs: An introduction to item response theory, scale linking, and score equating. *Research & Practice in Assessment, 8*(1), 26–39.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2004). *A brief introduction to evidence centered design. Technical Report CSE Report 632*. Los Angeles: The National Center for Research on Evaluation, Standards, and Student Testing.

Mitrovic, A. (2012). Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction, 22*, 39–72.

Mitrovic, A. & Weerasinghe, A. (2009). Revisiting ill-definedness and the consequences for ITSs. *14th International Conference on Artificial Intelligence in Education*, pp. 375–382.

Mitrovic, A., Martin, B., & Mayo, M. (2002). Using evaluation to shape its design: Results and experiences with sql-tutor. *User Modeling and User-Adapted Interaction, 12*(2–3), 243–279.

Mitrovic, A., Koedinger, K. R., & Martin, B. (2003). A comparative analysis of cognitive tutoring and constraint-based modeling, *Proceedings of the Ninth International Conference on User Modeling*, pp. 313–322.

Mitrovic, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., & Holland, J. (2006). Authoring Constraint-Based Tutors in ASPIRE. *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems*, pp. 41–50.

Mitrovic, A., Martin, B., & Suraweera, P. (2007). Intelligent Tutors for All: The Constraint-Based Approach. *IEEE Intelligent Systems, 22*, 38–45.

Ohlsson, S. (1994). Constraint-based student modeling. *Student Modeling: the Key to Individualized Knowledge-based Instruction*, pp. 167–189.

Ohlsson, S. (1996). Learning from performance errors. *Psychological Review, 103*(2), 241–262.

Ohlsson, S., & Mitrovic, A. (2006). Constraint-based knowledge representation for individualized instruction. *Computer Science and Information Systems, 3*, 1–22.

Pardos, Z. A., & Heffernan, N. T. (2011). KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Konstan, J., Conejo, R., Marzo, J.L., & Oliver, N. (eds.) *Proceedings of the 19th International Conference on User Modeling, Adaptation and Personalization*. Lecture Notes in Computer Science, vol. 6787, pp. 243–254

Pavlik, P. I., Cen, H. & Koedinger, K. R. (2009). Performance factors analysis - a new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pp. 531–538.

Sweller, J., van Merrienboer, J., & Pass, F. (1998). Cognitive architecture and instructional design. *Educational Psychology Review, 10*(3), 251–296.

Thurstone, L. L. (1925). A method of scaling psychological and educational tests. *Journal of Educational Psychology, 16*, 433–451.