Knowledge-Based Systems 97 (2016) 11–23

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Data calibration for statistical-based assessment in constraint-based tutors

Jaime Gálvez[a], Eduardo Guzmán[a,*], Ricardo Conejo[a], Antonija Mitrovic[b], Moffat Mathews[b]

[a] E.T.S.I. Informática, Dpto. Lenguajes y Ciencias de la Computación, Universidad de Málaga, Bulevar Louis Pasteur, 35, Campus de Teatinos, 29071 Málaga, Spain

[b] Department of Computer Science and Software Engineering, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand

ARTICLE INFO

Article history:
Received 28 September 2015
Revised 14 January 2016
Accepted 18 January 2016
Available online 25 January 2016

Keywords:
Learning analytics
Assessment
Constraint-Based Modeling
Intelligent Tutoring Systems
Item Response Theory

ABSTRACT

Intelligent Tutoring Systems (ITSs) are one of a wide range of learning environments, where the main activity is problem solving. One of the most successful approaches for implementing ITSs is Constraint-Based Modeling (CBM). Constraint-based tutors have been successfully used as drill-and-practice environments for learning. More recently CBM tutors have been complemented with a model derived from the field of Psychometrics. The goal of this synergy is to provide CBM tutors with a data-driven and sound mechanism of assessment, which mainly consists in applying the principles of Item Response Theory (IRT). The result of this synergy is, therefore, a formal approach that allows carrying out assessments of performance on problem solving tasks. Several previous studies were conducted proving the validity and utility of this combined approach with small groups of students, in short periods of time and using systems designed specifically for assessment purposes. In this paper, the approach has been extended and adapted to deal with a large set of students who used an ITS over a long period of time. The main research questions addressed in this paper are: (1) Which IRT models are more suitable to be used in a constrained-based tutor? (2) Can data collected from the ITS be used as a source for calibrating the constraints characteristic curves? (3) Which is the best strategy to assemble data for calibration? To answer these questions, we have analyzed three years of data from SQL-Tutor.

1. Introduction

Intelligent Tutoring Systems (ITSs) are probably the most well-known product of the Artificial Intelligence in Education (AIED) research community. ITSs are environments that help student learn a subject matter. To do that, they use a knowledge base that is comprised of a student model and a domain model, modeling what the student knows and what to teach, respectively. The teaching process of an ITS consists of consulting the knowledge base and adapting the content and tutorial actions according to the student model. This behavior tries to mimic an expert human teacher who adapts the process to every individual student. Perhaps the most extended interaction pattern an ITS provides is an environment where students can solve problems belonging to certain domain matter. According to Jonassen [18], "*most educators agree that problem solving is among the most meaningful and important kinds of learning and thinking*". A problem exists when a problem solver has a goal but does not know how to reach it. Problem solving is a mental activity aimed at finding a solution to a certain problem [3]. The challenge of solving a problem forces students to build models through a process of understanding, exploring and interacting with the world, developing several branches of science at all levels of education [30]. Thus, problem solving entails cognitive processing with the goal of transforming a given situation into a desired scenario when no obvious method of solution is available to the problem solver [21]. According to Mayer [22] problem solving expertise can be decomposed into four components:

1 *Problem translation*, where the student transforms the problem stem into an internal mental representation.
2 *Problem integration*, a mental model of the situation described in the problem stem is constructed.
3 *Solution planning*, where the strategy to solve the problem is determined, i.e. the steps to take in order to solve the problem. This component requires the student to apply his/her procedural knowledge.
4 *Solution execution*, that is, the previous plan is applied to solve the problem.

Constraint-Based Modeling (CBM) [39] is one of the most popular approaches for developing ITSs [8,43]. Its effectiveness as an

[*] Corresponding author. Tel.: +34 952137146.
E-mail address: guzman@lcc.uma.es (E. Guzmán).

http://dx.doi.org/10.1016/j.knosys.2016.01.024
0950-7051/© 2016 Elsevier B.V. All rights reserved.

Nomenclature

ITS:	Intelligent Tutoring System
AIED:	Artificial Intelligence in Education
CBM:	Constraint-Based Modeling
IRT:	Item Response Theory
ECD:	Evidence Centered Design
ICC:	Item Characteristic Curve
BN:	Bayesian Network
CCC:	Constraint Characteristic Curve

instructional methodology has been proved in a range of tutors and studies performed over 15 years [33,35,37,38]. A characteristic that makes it a very attractive approach is its ability to be applied in a tutoring system easily since it does not require a complex architecture. Furthermore, it does not require identifying all possible steps a student could take to reach a solution to a problem. Instead, it only requires the identification of domain principles (represented as constraints) that no solution should violate.

Educational assessment characterizes aspects of student knowledge, skill, abilities, or other attributes. For this characterization it makes inferences from the observation of what they say, do, or make in certain kinds of situations [5]. Furthermore, educational assessment provides at least three different uses in instructional improvement [3]: first, results obtained through assessment motivate students and educational staff to achieve the academic goals set by policy makers. In addition, it represents a way of helping teachers to plan or revise their pedagogical strategies. Finally, assessment can be used to help stimulate deep understanding. The use of computers in testing is extensive nowadays. In the area of problem solving, however, there is still an enormous range of opportunities to explore [3,52]. Problem solving activities require students to apply their knowledge in constructing a solution to a certain situation [23]. One of the most recognized assessment techniques is Item Response Theory (IRT), which gave rise to a set of different models with different assumptions (see next section).

In our previous work [14,15] we made a first proposal of a model of assessment combining CBM with IRT. This proposal can also be seen as an implementation of the Evidence Centered Design (ECD) framework [1,29,41], which focuses on providing a generic methodology to perform assessments of problem solving. This synergy between the AIED and psychometric mechanisms opens the door to enhancing ITSs with new methods to perform automatic assessment of tasks that, if carried out by a human expert, would be highly difficult and prone to subjectivity. As will be explained later, the utilization of IRT makes it possible to apply new formal psychometric methods in CBM that were not possible before. In the same way, some of the fundamentals of CBM extend the typical use of IRT in testing environments, where theoretical concepts are assessed, to ITS, which requires applying practical knowledge to solve a problem.

Initially, in order to explore the validity of the approach for assessment purposes, two educational systems were developed and tested with undergraduate students of the University of Malaga in Spain [13–15]. Although the knowledge base of these ITSs was developed in well-defined domains, according to the classification made by Mitrovic and Weerasinghe [36], the tasks involved were completely different. In the first system, focused on the Simplex algorithm for mathematical optimization, the number of constraints was small and the tasks were well-defined (i.e. those tasks for which the process of solving them is known). On the other hand, the second system, focused on teaching fundamentals of Object Oriented Programming, had a relatively large number of constraints and the tasks were ill defined with a complex solution procedure (having more than one solution or many ways to achieve it).

Initial results obtained using CBM and IRT showed that the methodology was feasible and promising in these types of domains. Nevertheless, the experiments were carried out in systems constructed for assessment purposes, with a small group of students, using a particular IRT model and strictly following the restrictions imposed by the IRT to guarantee valid assessment results under this theory. To the contrary, the most successful CBM-based systems have been used mainly for learning purposes in drill-and-practice environments. That means that a student is allowed to solve the same problems several times which leads to the violation of the IRT models assumed hypotheses (i.e. student knowledge is constant during a session). This difference makes it necessary to explore the scalability and validity of the existing models based on the combination of IRT with CBM in tutoring systems used for learning purposes and with a large number of students.

The research carried out in this paper tries to cover the aforementioned problems by extending the existing methodology (explained in detail in the following sections) and performing a study with a larger dataset obtained over three years of use of the SQL-Tutor [34]. The aims of the study are: (1) to define an appropriate methodology to accommodate IRT models to constraint-based tutors; (2) to determine the most appropriate IRT models in this case; and (3) to explore different strategies for grouping and filtering existing ITS data to be used for the IRT calibration process. The advantages of using this approach are that it provides a data-driven technique that does not require heuristic knowledge. The resulting ITS would be adjusted by standard statistical calibration procedures that are not biased with the designer subjectivity.

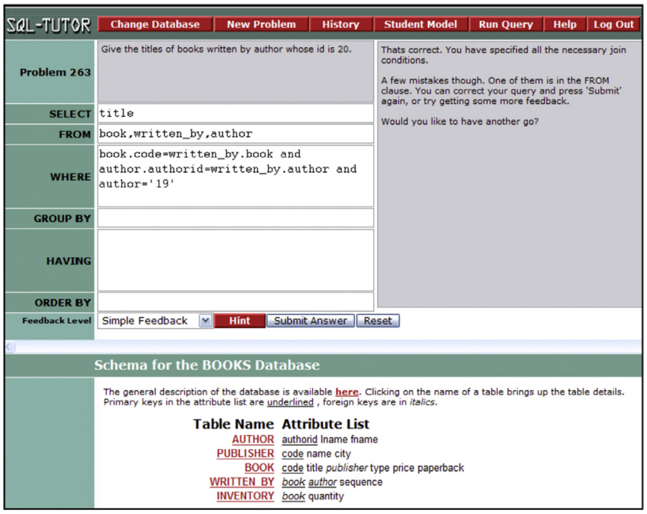
The paper is structured as follows: Section 2 presents the theoretical background needed to understand both the model and the calibration strategies presented in this paper. Section 3 describes the related work in the field of AIED. Section 4 is devoted to a formalization of our assessment model and a generalization of that model to be used for ITS under the Evidence-Centered Design framework; it also outlines the drawbacks of the early approach. Section 5 proposes a new methodology to overcome the limitations of our proposal with several strategies that can be performed in the process of calibration. Section 6 describes the experiments and the methodological issues and Section 7 presents and discusses the results. Finally, conclusions are summarized in Section 8.

2. Theoretical background

The approach for assessment in ITSs is based on two main pillars, corresponding to the two methodologies already mentioned: CBM for modeling the ITS domain, and the IRT for assessing the student's knowledge in terms of the evidence provided by him/her while solving problems. Both techniques are summarized here. Moreover, the system used in this paper, i.e. SQL-Tutor, is also described briefly.

2.1. Constraint-Based Modeling

The first element of the methodology is the CBM paradigm for building ITSs, which will be the instrument through which students' evidence is gathered. CBM is based on Ohlsson's theory of learning from performance errors [39,40], according to which incomplete or incorrect student's knowledge can be used within an ITS to provide guidance. This faulty knowledge is detected using constraints, which are the key element of CBM. Constraints are principles that must be followed by all correct solutions in the given instructional domain. If the student's solution violates any constraints, it is incorrect and the system provides the student with the appropriate feedback for remediation. Each constraint



Problem 263

Give the titles of books written by author whose id is 20.

SELECT title

FROM book,written_by,author

WHERE book.code=written_by.book and author.authorid=written_by.author and author='19'

Feedback Level: Simple Feedback

Thats correct. You have specified all the necessary join conditions.

A few mistakes though. One of them is in the FROM clause. You can correct your query and press 'Submit' again, or try getting some more feedback.

Would you like to have another go?

Schema for the BOOKS Database

The general description of the database is available here. Clicking on the name of a table brings up the table details. Primary keys in the attribute list are underlined , foreign keys are in *italics*.

Table Name	Attribute List
AUTHOR	authorid lname fname
PUBLISHER	code name city
BOOK	code title *publisher* type price paperback
WRITTEN_BY	*book* *author* sequence
INVENTORY	*book* quantity

Fig. 1. The SQL-Tutor interface.

consists therefore of an ordered pair (C_r, C_s), where C_r is the relevance condition and C_s is the satisfaction condition [33]:

If < *relevance condition* C_r > is true,

then < *satisfaction condition* C_s > had better also be true.

The application of CBM is very simple, since only an inference engine and the appropriate representation of the solution are required [31]. Accordingly, once the student has finished solving a problem (or it can also be done before by student demand), constraints are checked against the student's solution using simple pattern matching. Constraints are only applied to solutions for which they are relevant (as determined by the relevance condition of each constraint). The satisfaction condition of a relevant constraint specifies properties that the solution must fulfill to be correct. The set of constraints and problems that can be presented to students form the domain model of a particular tutor. The performance of a student with respect to the constraints, i.e., the list of violated and satisfied constraints in each solution take part of his/her student model.

2.2. SQL-Tutor

In this paper, we have used data from one of the most popular and successful constraint-based tutors, SQL-Tutor [34]. Although its main source of users comes from the students enrolled in database courses at the University of Canterbury in New Zealand, it is available worldwide via the DatabasePlace portal established by Addison Wesley,[1] which uses SQL-Tutor and two other tutors developed in the databases domain [32].

SQL-Tutor teaches SQL queries, which is the dominant relational database query language. It is designed to help undergraduate students with their difficulties mastering the subject. Although SQL is a simple and well-structured language, students find it difficult to learn due to the advanced concepts and cognitive overload [45] associated with this type of problem, which is a result of having to keep in mind many details involved in the problem that is being solved.

The interface of SQL-Tutor reduces the working memory load by displaying the database schema and other information related to the problem (see Fig.1). Without this information, the student would have to keep in mind the structure of the database or handle it by other means. Besides, the system presents the parts of the solution, simplifying the problem in different subgoals, each one associated with the building of a particular component.

The correctness of a student's solution can be verified by submitting it to the system. Incomplete solutions can be submitted too. The system compares the student's solution to the constraints. SQL-Tutor's domain model is comprised of a huge set of constraints, with more than 700 defined so far. This can give the reader an idea about the difference in magnitude between the data that can be obtained with this system, with respect to the systems used in existing studies, where the most complex domain was comprised of 87 constraints and the simplest had 18. Examples of constraints are shown in Fig. 2.

The violations and satisfactions of the constraints are used to inform the students about their mistakes. The system provides feedback in increasing levels of detail, starting from one that gives little information to one that gives the complete solution [20]. The history of use of each constraint is stored in the student model, showing for each attempt whether the constraint was used correctly or whether it was violated.

Simultaneously with the process described above, the system records all relevant activities of each student in a log file. This includes all the results that affect the student model and the scaffolding information. This log file containing qualitative information about the students has been the source of evidence used in the research presented in this paper.

2.3. The Item Response Theory

The second pillar of our assessment model a well-founded technique specifically developed for assessment, i.e. the IRT [49]. This theory assumes that a latent trait (i.e. the student knowledge level) can be inferred from the student's answers to independent questions or items, which provides evidence based on conditional probabilities named the Item Characteristic Curve (ICC) [17]. Its main advantage in comparison with other assessment techniques is the invariance of measurement. This means that the assessment score is independent of the instrument of measure being used and, thus, the same score would be obtained in any test taken [16].

The ICC, which is probably the most important concept in the IRT, models the probability of answering a question correctly given the student knowledge. Fig. 3 illustrates the shape of the ICC. As can be seen, the greater the knowledge value (x axis), the higher the probability of giving a correct response (y axis). There are different IRT models based on different ICC functions. This figure contains what is probably the most popular function that implements the ICC, i.e. the 3 Parameters Logistic (3PL), which is also depicted in the equation below:

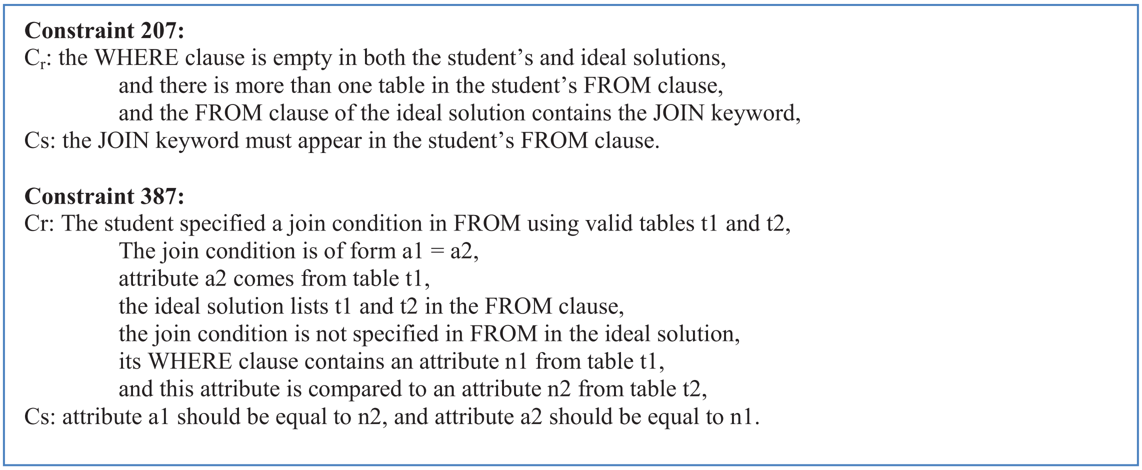
$$P(u_i = 1|\theta) = c_i + (1 - c_i)\frac{1}{1 + e^{-1.7a_i(\theta - b_i)}} \qquad (1)$$

Here, $P(u_i = 1|\theta)$ represents the probability of correctly answering the item i, given a student's knowledge level θ within the interval ($-\infty$,…,$+\infty$). The correctness of the question is represented with $u_i = 1$, otherwise 0 would be used to reflect a wrong state. The other elements in the equation are the three parameters characteristic of the 3PL function:

- a_i is called *discrimination factor* and is a value proportional to the slope of the curve. The greater this value, the higher the distinction between different student's knowledge levels.
- b_i, also called *difficulty*, is the value of θ for which the probability of answering correctly is the same as answering wrongly.
- The last parameter, c_i, is the *guessing factor* and represents the probability of a student without knowledge answering correctly.

Only those models whose items can be assessed as correct or incorrect, i.e. the dichotomous models, are considered here, such

[1] http://www.aw.com/databaseplacedemo/sqltutor.html.



Constraint 207:

C_r: the WHERE clause is empty in both the student's and ideal solutions,
and there is more than one table in the student's FROM clause,
and the FROM clause of the ideal solution contains the JOIN keyword,
Cs: the JOIN keyword must appear in the student's FROM clause.

Constraint 387:

Cr: The student specified a join condition in FROM using valid tables t1 and t2,
The join condition is of form a1 = a2,
attribute a2 comes from table t1,
the ideal solution lists t1 and t2 in the FROM clause,
the join condition is not specified in FROM in the ideal solution,
its WHERE clause contains an attribute n1 from table t1,
and this attribute is compared to an attribute n2 from table t2,
Cs: attribute a1 should be equal to n2, and attribute a2 should be equal to n1.

Fig. 2. Examples of constraints in SQL-Tutor.

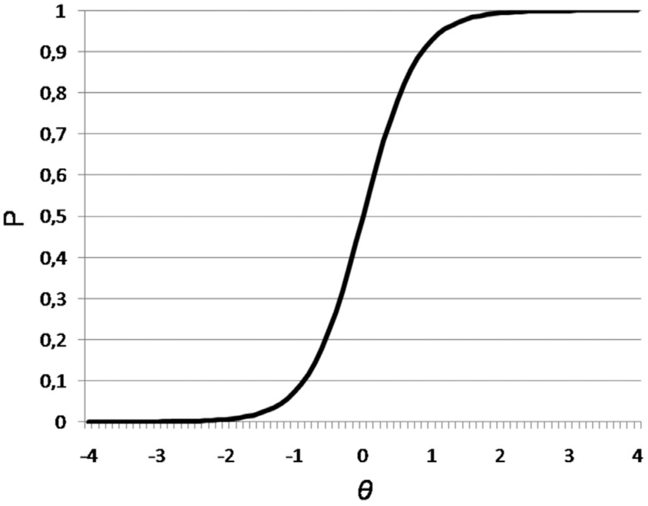


Fig. 3. The shape of an ICC under the 3PL model.

as the Two Parameters Logistic (2PL) or the One Parameter Logistic (1PL). Both of them are simplifications of the 3PL function: the 2PL is equivalent to the 3PL but the guessing parameter would be 0, and the 1PL is equivalent to the 2PL but fixing the discrimination parameter to a given value, i.e. $a_i = 1$. However, there are other approaches, e.g., the polytomous models, where more than two answers are allowed and therefore partial credit to items can be given [17]. This initial decision is congruent because constraints are dichotomous.

Using the ICCs, and assuming (1) item independence; and (2) constant knowledge throughout the session, the knowledge of the jth student θ_j can be computed as shown is equation:

$$P(\theta_j) = \prod_{i=1}^{n} P(u_i = 1|\theta_j)^{u_{ij}} \left[1 - P(u_i = 1|\theta_j)\right]^{1-u_{ij}} \quad (2)$$

where $P(\theta_j)$ is the jth student knowledge distribution; n is the number of items administered to the student; $u_{ij} = 1$ indicates that the jth student's answer to item i was correct, otherwise $u_{ij} = 0$.

The likelihood function of a given set of response patterns is therefore:

$$L(u|a_i, b_i, c_i, \theta_j) = \prod_{j=1}^{N} \prod_{i=1}^{n} P(u_i=1|\theta_j)^{u_{ij}} \left[1-P(u_i=1|\theta_j)\right]^{1-u_{ij}} \quad (3)$$

where N is the total number of students.

There are different techniques for estimating the model parameters a_i, b_i, c_i and the students' knowledge θ_j that maximizes this function. One of them is the Marginal Maximum likelihood (MML). This process is known as calibration and is carried out with the help of the computer program Multilog [48].

In order to compare the goodness of fit of two different models, with a different number of parameters, the ratio of the likelihood function can be used. The test statistic is twice the difference in these log-likelihoods:

$$D = -2\ln\left(\frac{L_1}{L_2}\right) = \; -2\ln(L_1) + 2\ln(L_2) \approx \chi^2(g) \quad (4)$$

where g is the degree of freedom, which is computed as the difference in the number of parameters of the two models. Multilog output includes the negative-twice-log-likelihood value for each model calibration. A model with more parameters will always fit at least as well (have an equal or lower negative-twice-log-likelihood). Whether it fits significantly better and should thus be preferred is determined by deriving the probability or p-Value of the difference D.

3. Related work

There are three outstanding approaches for developing ITSs: cognitive tutors, Bayesian Networks (BNs) and CBM. Cognitive tutors are learning environments based on the ACT-R theory of cognition [2]. That theory makes a distinction between declarative and procedural knowledge. The first one involves factual knowledge, whereas the second is based on production rules which enable students to solve problems. Cognitive tutors include their own mechanism to estimate the student's knowledge during the learning process, i.e. Bayesian Knowledge Tracing [10]. It models the knowledge through hidden Markov models where binary values are assumed and give as a result short-term student models, i.e. models oriented to adapt the instructional process according to the estimations obtained during that process.

BNs are probably the most widespread approaches that have been used for modeling student knowledge while solving a complex task [12,42]. They are graphical modeling tools that have been successfully applied in different application contexts [26]. These networks model the probability of a student mastering a specific knowledge component in terms of the sequence of responses given to previous elements of a task [12]. BNs have been applied in intelligent tutoring systems to represent student knowledge, e.g. [7,9,25,44,46,50,51]. BNs can also be combined with other techniques such as machine learning [4]. When used

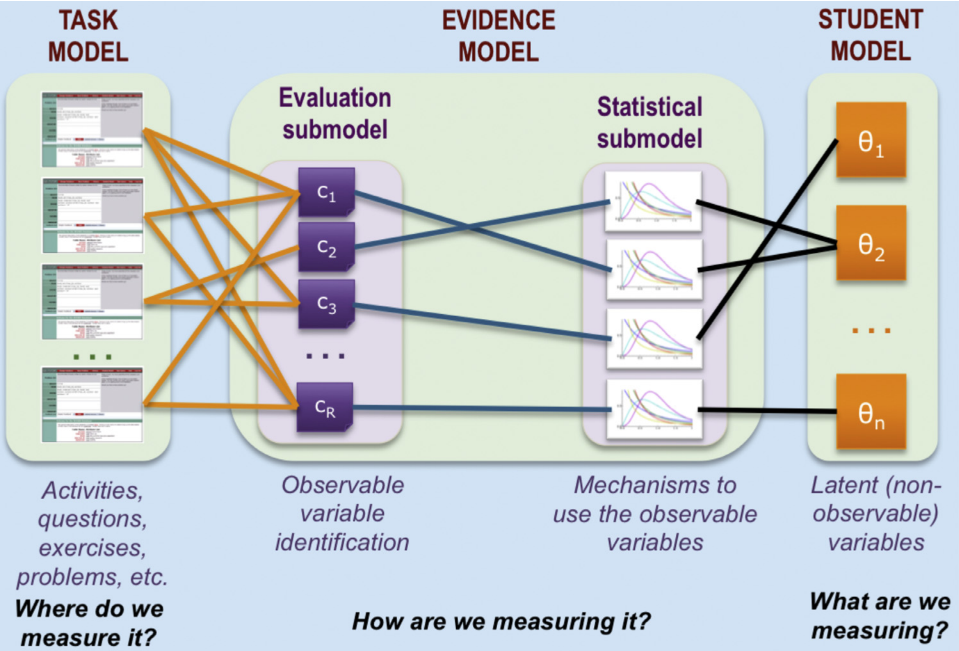


Fig. 4. Conceptual assessment framework.

for assessment purposes, nodes of a BN can represent different components of an individual such as knowledge, misconceptions, emotions, learning styles, motivation, goals, etc. [8]. However, the main drawbacks of BNs is the way of constructing the networks that could affect to the results obtained, and the calibration of the conditional probabilities, which is a complex process.

In the existing literature we have not found any other formal methodology applied in CBM to assess students. Although in [24] BNs were used to model the student by estimating the probability of mastering a constraint, the estimates were not used as a medium to get the students' level, but to provide them with the most appropriate instructional action. Even looking more generally in the field, at the level of assessment in ITSs, we were unable to find a well-founded approach that, using student's interaction with the system, automatically generates well-founded assessments.

More recently, Davier and Halpin [12] proposed a framework for the assessment of cognitive skills in problem-solving tasks solved collaboratively. They also proposed several statistical approaches to model the data collected from collaborative interactions, where they tried to measure separately the contribution of each student to the final solution of the problem.

4. An assessment model for problem solving environments

Even though testing is the most common approach for assessment in computer-based systems, there are some domains (especially those involving procedural tasks) where this evaluation mechanism does not seem to be the most suitable. Several authors such as [6] have pointed out that in any learning system designed to emulate professional practice the assessment should be performance based. Our proposal here is directly aligned with that claim: students' knowledge acquired in problem solving environments should be measured in the same way, i.e. using a few problems instead of forcing the students to take a test composed of a large number of questions about the knowledge required to solve those problems.

The goal of our assessment model is to provide a framework for building assessment systems based on constraint-based tutors powered by IRT models. Consequently, this proposal is the result of combining two different lines of research, i.e. CBM and IRT-based assessment, into a single environment able to be used both for assessment and for learning purposes. As a result of this combination, a constraint-based tutor would be also able to perform a formal and quantitative estimation of the student knowledge.

Our proposal can be framed under the *Evidence-Centered Design* (ECD) methodology, which is a guideline for designing, producing and delivering educational assessments [28,29]. It incorporates representations of what a student knows and does not know, in terms of the results of his/her interaction performance (evidence) with assessment tasks. According to Behrens et al. [5], "*ECD framework provides terminology and representations for layers at which fundamental entities, relationships, and activities continually appear in assessments of all kinds*". Knowledge representations, workflows, and communications are organized in terms of layers [27]. Five layers can be identified in ECD which are summarized below together with the way in which they have been particularized for our proposal:

- *Domain analysis*, where relevant information about domains is gathered, i.e., concepts, terminology, tools, knowledge representations, etc. In our case it consists of identifying the concepts, skill, etc. involved in each problem and the constraints that characterize the domain.
- *Domain modeling*, where the results of the previous layer are represented in a model, in terms of assessment argument. For our proposal, knowledge, skills and abilities are identified. They will be measured in the student model. Additionally, observable knowledge evidences are collected and, thus, the set of constraints identified during the analysis will be included in the problems which will take part of the task model.
- *Conceptual assessment framework:* Structures of the assessment model are designed (see Fig. 4). Here student observable knowledge evidences (on left-hand side of the figure) are related to non-observable features such as the student knowledge (right-hand side of the figure). The *Student model* will consist of probability distributions containing estimations of the student knowledge, skills and abilities identified in the domain modeling stage. In Fig. 4 these estimations are repre-

sented as $\theta_1, \theta_2, \ldots, \theta_n$. These unobservable variables are linked to the observable evidences through an *evidence model*, able to transform those knowledge evidences into updates of the student model. These observable evidences are provided by the *task model*, which comprises the set of exercises or problems the student has to solve (on the left-hand side of Fig. 4). More concretely, in our proposal the task model consists of the set of problems provided by a CBM tutor such as SQL-Tutor. The evidence model uses the evidence provided by the CBM-based problems and an IRT model is applied to them. Inside the *evidence model*, two submodels can be found: the *evaluation submodel* identifies the observable elements in the task model, which will be used to perform the assessment. The *statistical submodel* is responsible for the transformation of the observable evidence into updates of the student model. The next section will describe this evidence model in detail.

- *Assessment implementation:* The model generated as a result of the previous layers is implemented and calibrated. As mentioned, calibration is a previous stage that needs to be done before the assessment.
- *Assessment delivery:* Finally, the result of all previous layers is compiled and used in an empirical environment to assess the performance of students.

4.1. The evidence model

Test items in assessment are usually scored dichotomously, i.e. either as correct or incorrect. However, problems in constraint-based tutors, from a psychometric perspective, can be seen as what is called *constructed-response questions* [19]. The performance of students on such problems is difficult to evaluate, as they require different types of knowledge, skills, or abilities to be applied (e.g. the design of a laboratory experiment, solving a mathematical problem, writing a schema summarizing a text, etc.). Assessment of complex tasks requires more sophisticated mechanisms taking into account all the knowledge needed to find the solution. The final solution in these kinds of tasks is not thus a good indicator of the students' knowledge level in the subject matter. When a human tutor evaluates the student's performance on a complex task, he/she not only checks whether or not the solution is correct, but also explores how the students accomplished the process of solving the tasks. That is, for the evaluation of that task, several evidences are taken into account in order to compute the score in it.

In order to overcome the limitations that constructed-response questions usually have when they are treated like multiple-choice questions from the assessment point of view, in our proposal those complex tasks are considered a source of multiple student knowledge (or un-knowledge) evidence. In constraint-based tutors each problem is linked with a set of constraints representing domain principles. As a result, students, while solving a problem, are generating evidence through the constraints they violate or satisfy. We use such evidence to compute the student knowledge applying an IRT-based assessment model. The set of constraints constitutes the evaluation submodel. Accordingly, in our evaluation model constraints are treated as IRT-based items. Constraints and items have the same nature since they provide evidence on the student's declarative knowledge: in IRT, a test item provides evidence about a domain concept being assessed. In the same way, a constraint provides evidence about a domain principle while the student is solving a problem. Both constraints and items take two values that represent the student's performance, which can be used as a source of evidence to estimate the knowledge level. When a student is solving a problem, there will be a set of relevant constraints, that is, those constraints that could be violated in the problem. As a result, once the student has solved a problem, we can get the set of constraints (which are relevant for that problem) that were (or not) violated.

In the statistical submodel each constraint c_j will have its own characteristic curve, $P(c_j|\theta)$, representing the probability of violating it given the student knowledge level θ. Those characteristic curves are called *Constraint Characteristic Curves* (CCCs) in analogy to the IRT ICCs, and through them the kth student knowledge level $P(\theta_k|\phi, \tau)$ can be computed as can be seen with the equation:

$$P(\theta_k|\phi, \tau) = \prod_{i=1}^{m} \prod_{j=1}^{n} \left[P\left(c_j|\theta_k\right)^{f_{ij}} \left(1 - P\left(c_j|\theta_k\right)\right)^{1-f_{ij}} \right]^{r_{ij}} \tag{5}$$

In Eq. (5), $\phi = p_1, p_2, \ldots, p_m$ represents the set of m problems solved by the kth student and $\tau = c_1, c_2, \ldots, c_n$, the set of all domain constraints. Note that the same constraint can appear in different problems. Accordingly, r_{ij} indicates whether or not the jth constraint is relevant in the ith problem. Moreover, $f_{ij} = 1$ indicates that the constraint c_j was violated in the problem p_i. Otherwise, f_{ij} is zero. The student knowledge is expressed as a probability distribution computed as a product of CCCs or their inverse depending on whether or not the constraint was violated.

5. Constraint characteristic curves calibration

Characteristic curves need to be calibrated before being used for assessment purposes. As a result of that process, the parameters of the characteristic curves are calculated. In testing environments, the original calibration process is done using student performance results. More concretely, the value of correction or mistake for every question and for each student from a sample is needed. The data needed can be represented with a matrix reflecting the performance of the student, henceforth called the *Performance Matrix*. Each row of this matrix is the data of a single student and each column is the result of a student for all the questions. For example, the element e_{ij} of the matrix would be the result for the student j in the question i. The elements can take three values: 1 to represent positive result (answered correctly); 0 to indicate a negative result (an incorrect answer); and another fixed value to indicate that the question has not been presented to the student.

The process of calibrating can be done using the performance matrix as input for IRT software such as, for instance, Multilog [48]. Nevertheless, in the case of the CBM approach, setting up the values of the elements in the matrix needs to take into account some principles in order to produce a valid model. These key principles arise from the IRT assumptions that must be satisfied in order to produce a valid model and estimates:

1) *Local independence* of the items being calibrated, meaning that one item should not provide any information a student could use to correctly answer another item.
2) *Constant knowledge*, which establishes that during the test, the measured latent trait does not change. This hypothesis implies that the knowledge should be the same for the entire assessment session, i.e. no learning could occur meanwhile.

The previous procedure of estimating characteristic curves can be applied to calibrate CCCs. In this case, however, the input of this calibration process is the performance of a student population who previously solved the set of problems. The performance matrix is composed, therefore, of a row for each student and a column for each constraint. The three possible values would have the same meaning: 1 for a positive result (satisfying the constraint), 0 for a negative result (violating the constraint), and another fixed value for a constraint that has not been relevant to the student's solution. The calibration outcome is the set of CCCs. As mentioned, each one of these curves models the probability of violating a constraint given a certain level of knowledge. The shape of a typical CCC is the exact opposite of an ICC (see Fig. 3). Therefore, it would be a monotonically decreasing function since the higher the knowledge, the lower the probability of violating the constraint.

Regarding the IRT assumptions that have to be fulfilled before performing the calibration, the first one, i.e. the local independence, is satisfied by the CBM itself since the constraints must be basic and exclusive principles. Nevertheless, the second assumption could be conflicting within ITSs since these types of systems are made for learning purposes and, in that case, the student's knowledge usually changes. In the rest of this section, we will explore three different strategies (i.e. the "constant knowledge sessions", the "first time relevant", and the "problem grouping") to approach calibration when available student data do not fulfill the requirement of constant student knowledge. Finally, an example will be shown to contribute to a better understanding of those three criteria.

5.1. *The* "Constant Knowledge session" *approach*

In our previous work [14,15] good results were achieved by applying IRT to CBM in problem solving assessment environments. However, here, we want to go further and design a procedure for calibrating the CCCs for constraint-based tutors. The challenge is therefore to be able to calibrate the CCCs for systems aimed not only at assessment, but also at learning.

To tackle the above-mentioned issue, we designed a new strategy to build the performance matrix by redefining the concepts of "session" and "student". Normally, a session takes place when the student logs into the ITS, carries out some or activities and then logs out. If the student's activities in consecutive sessions are grouped considering those sessions close enough in time, we could have windows of activity where the knowledge between sessions could be assumed to be constant or not significantly different. This concept is what we call a *Constant Knowledge session* (CK-session). The time separating any two consecutive sessions in a CK-session should not be higher than a certain threshold. It can be stated formally in the following way: Let a_{mi} be the moment the last student action happened in the *i*th session (S_i); $a_{0(i+1)}$ the moment the first action occurred in the $(i+1)$th session (S_{i+1}); and T_{CK} a fixed threshold that represents a period of time where it can be assumed that the knowledge has not changed. If $(a_{0(i+1)} - a_{mi}) < T_{CK}$ then, S_i and $S_{(i+1)}$ will belong to the same CK-session.

All CK-sessions of a student must be taken into account in the CCC calibration, since these sessions provide information about different sets of constraints. However, each CK-session represents a different knowledge state of the student, as stated before, and considering the whole set of evidence for a student would thus violate the IRT assumptions. This problem can be tackled by splitting each different CK-session of a student into separate sessions of different *virtual students*. In this way, the set of a student's CK-sessions could be turned into a larger set of virtual students, each one having a different knowledge state. It is important to note that this strategy avoids *inter-session learning*, but it is still necessary to bear in mind the intra-session learning. The *intra-session learning* can be avoided by using the students' evidence of a constraint only the first time it was relevant and avoiding the learning provided by feedback inside the CK-session.

5.2. *The* "first time relevant" *approach*

The problem of constant knowledge can be dealt by selecting in the calibration only those values representing the students' performance that did not result in learning gain. Identifying such values is relatively easy in those cases it was used evidence from CBM tutors that were designed for assessment purposes. That "first time relevant" approach takes as evidence only the student performance the first time the constraint is relevant for the student. This criterion is equivalent to setting the T_{CK} to be greater than the whole period where the evidence is being taken. Therefore, we only used the result of a constraint the first time it could be (or not) violated, since the principle it models makes sense in the current problem state. For instance, in the domain of fraction addition, constraints on computing the least common multiple make sense only when the student is calculating it.

By considering the first time a constraint is relevant, we are only taking into account the student's prior knowledge state, i.e. the knowledge before learning. Otherwise, if we would also consider what happened the *n*th time (where $n > 1$) a constraint was relevant, we would not be taking into account the fact that a previous violation of the constraint could have resulted in feedback which could modify the student's knowledge state associated with that constraint. In this way, the performance matrix used for calibration in the existing experiments was formed by filtering the values for repeated constraints.

Let us take for example SQL-Tutor. In this CBM system, there is neither any restriction about the number of attempts per problem, nor any imposition on the sequence of problems to be solved. Therefore, the students can have many sessions with the tutor, whenever they want, and solve as many problems per session as they like. This means that a constraint can be relevant at different times for each student and multiple times, each one reflecting different knowledge stages. Using this calibration approach of the first time relevant in systems where students have large sessions, discards data associated with constraints that are not relevant for the first time but, however, could be associated with new states of the student's knowledge. Missing these data involves redesigning the existing strategy to take into account the student knowledge evolution that occurs over long periods of usage and, in general, in any tutoring system. This problem can be solved by combining this approach with the CK-session approach.

5.3. *The* "problem grouping" *approach*

The "problems grouping" criterion consists in grouping the evidences by problems, which means that consecutive attempts of a student to solve a problem are considered to be in the same CK-session and, thus, conforming to a virtual student. Although this criterion has a variable value of T_{CK}, because between two different problems done by a student there is no fixed amount of time, we thought it would be interesting to make this distinction to assume knowledge changes only between problems.

5.4. *Data filtering*

Given that constraints are relevant for specific problems, the amount of evidence obtained for these constraints will depend on how often the problems are attempted by students. In domains with large constraint sets, such as SQL-Tutor, a high level of interaction between the students and the system is required in order to have a homogeneous amount of evidence per constraint. For this reason, some of the constraints will have a smaller amount of evidence than others, and, therefore, will produce less accurate calibration. Taking this into account, we have considered three filtering scenarios:

- *Scenario 1:* Full data set. This is the basic scenario where constraints that were not relevant during a given year, that is, those that were not included in any of the problems of that year, were discarded for the calibration process in that year.
- *Scenario 2:* Discarding constraints which are only rarely relevant. Constraints that were relevant for less than 10% of times they could have been relevant were discarded.

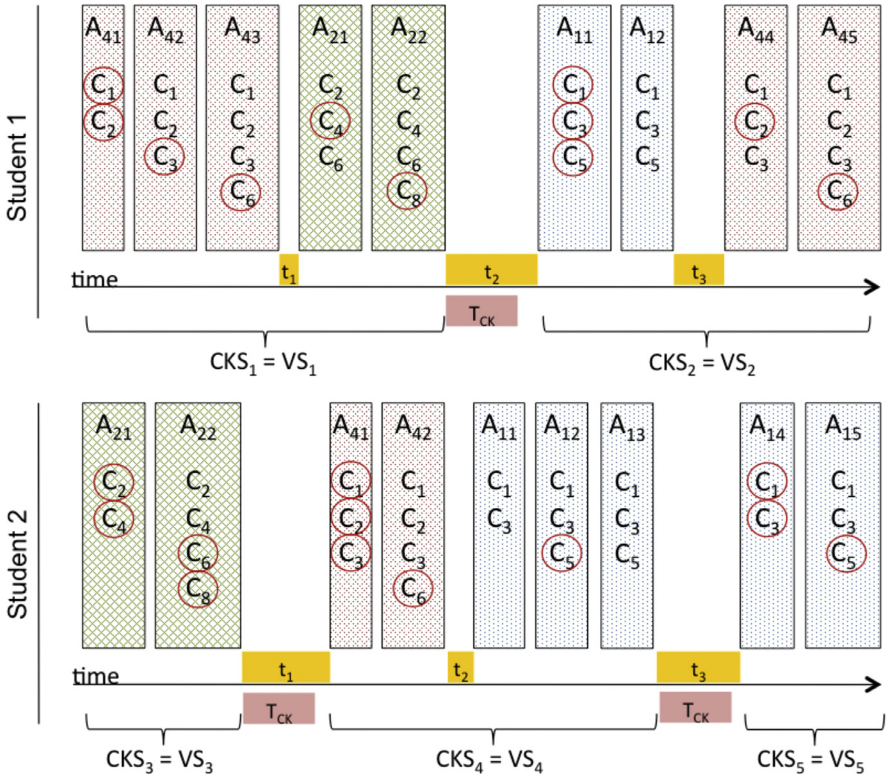

Student 1
A_{41} C_1 C_2
A_{42} C_1 C_2 C_3
A_{43} C_1 C_2 C_3 C_6
A_{21} C_2 C_4 C_6
A_{22} C_2 C_4 C_6 C_8
A_{11} C_1 C_3 C_5
A_{12} C_1 C_3 C_5
A_{44} C_1 C_2 C_3
A_{45} C_1 C_2 C_3 C_6
time
t_1
t_2
t_3
T_{CK}
$CKS_1 = VS_1$
$CKS_2 = VS_2$
Student 2
A_{21} C_2 C_4
A_{22} C_2 C_4 C_6 C_8
A_{41} C_1 C_2 C_3
A_{42} C_1 C_2 C_3 C_6
A_{11} C_1 C_3
A_{12} C_1 C_3 C_5
A_{13} C_1 C_3 C_5
A_{14} C_1 C_3
A_{15} C_1 C_3 C_5
time
t_1
t_2
t_3
T_{CK}
T_{CK}
$CKS_3 = VS_3$
$CKS_4 = VS_4$
$CKS_5 = VS_5$

Fig. 5. A graphical representation of two students' performances in an ITS.

- *Scenario 3:* Discarding constraints with low variability (almost always violated or always not violated). We also considered that it would be interesting to explore the effect of discarding not only constraints with a small amount of evidence, but also those that were usually violated or usually not violated by students when they were relevant to a problem. In this scenario we discarded the constraints that were violated less than 5% of the time, and those that were violated more than 95% of the time.

5.5. An example of use

To provide a better understanding of the proposed criteria, in Fig. 5 we introduce an example with a small set of eight constraints and two students. This figure shows a series of attempts, each represented by a rectangle labeled A_{ij}, meaning the attempted number j for problem i. Each attempt has a list of relevant constraints, which can be different for two attempts on the same problem, since the student could have added new elements in the submitted solution.

In this example, student 1 has made three attempts at problem 4; then, two attempts at problem 2; next, two attempts at problem 1; and again two more attempts at problem 4. The horizontal space between each pair of attempts represents the time elapsed between them. In this case, three significant spaces between the four problems solved by student 1 can be observed: t_1, t_2 and t_3. The performance matrices resulting from applying the different calibration approaches are represented in Fig. 6. The matrix corresponding to the CK-session approach is created by grouping the attempts which are not separated by more than a threshold T_{CK}. In the example presented in Fig. 5, for student 1, we can see that only t_2 is higher than the threshold value and, therefore, two CK-sessions can be considered (CKS_1 and CKS_2), each one representing a single session of two virtual students (VS_1 and VS_2). However, for student 2, both t_1 and t_3 are higher than the threshold and, as a result three virtual students are generated (VS_3, VS_4 and VS_5). Finally, in the figure we have circled those constraints that are relevant for the first time in a CK-session.

Note that it is possible for the time between two consecutive attempts a$_{ij}$ and $a_{i(j+\ 1)}$ to be greater than the time between two attempts in different (but consecutive) problems a_{ij} and a_{hk} (i.e., problem h attempted immediately after problem i). In that case, unless the student had closed the session for some reason, they are considered as still belonging to the same CK-session as, during this time, the student is supposed to be working with the system on a given attempt. For this reason, in the process of identifying CK-sessions from the data only pauses between different problems are considered.

Performance matrices corresponding to Fig. 5 according to the three criteria are given in Fig. 6. There, each column is associated with a constraint C_i and each row to a virtual student. Element e_{ij} in each matrix has an element A_{ap}, which represents that the performance result of constraint j for the virtual student i was taken from the attempt number a in the problem p. This result will be a binary value, 1 or 0, to represent the satisfaction or violation, or the character x when the constraint has not been relevant during the session.

Finally, the calibration is performed by applying some IRT method to the matrix obtained from any of the three approaches. The result will be the CCCs of all the constraints (more concretely, the parameters of the probabilistic function selected for modeling those curves), as well as the student knowledge estimation of those individuals whose data were used in the calibration process. In the study described in the next section, we have used Multilog software to infer the parameters associated with the logistic functions modeling the characteristic curves.

CK-Session

$$\begin{array}{ll|cccccccc} & & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\ \hline \text{Student 1} & CKS_1 = VS_1 & A_{41} & A_{41} & A_{42} & A_{21} & x & A_{43} & x & A_{22} \\ & CKS_2 = VS_2 & A_{11} & A_{44} & A_{11} & x & A_{11} & A_{45} & x & x \\ \text{Student 2} & CKS_3 = VS_3 & x & A_{21} & x & A_{21} & x & A_{22} & x & A_{22} \\ & CKS_4 = VS_4 & A_{41} & A_{41} & A_{41} & x & A_{12} & A_{42} & x & x \\ & CKS_5 = VS_5 & A_{14} & x & A_{14} & x & A_{15} & x & x & x \end{array}$$

1st time relevant

$$\begin{array}{ll|cccccccc} & & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\ \hline \text{Student 1} = & CKS_1 = VS_1 & A_{41} & A_{41} & A_{42} & A_{21} & A_{11} & A_{43} & x & A_{22} \\ \text{Student 2} = & CKS_2 = VS_2 & A_{41} & A_{21} & A_{41} & A_{21} & A_{12} & A_{22} & x & A_{22} \end{array}$$

Problems grouping

$$\begin{array}{ll|cccccccc} & & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\ \hline \text{Student 1} & CKS_1 = VS_1 & A_{41} & A_{41} & A_{42} & x & x & A_{43} & x & x \\ & CKS_2 = VS_2 & x & A_{21} & x & A_{21} & x & A_{21} & x & A_{22} \\ & CKS_3 = VS_3 & A_{11} & x & A_{11} & x & x & A_{11} & x & x \\ \text{Student 2} & CKS_4 = VS_4 & x & A_{21} & x & A_{21} & x & A_{22} & x & A_{22} \\ & CKS_5 = VS_5 & A_{41} & A_{41} & A_{41} & x & x & A_{42} & x & x \\ & CKS_6 = VS_6 & A_{11} & x & A_{11} & x & A_{12} & x & x & x \end{array}$$

Fig. 6. Performance matrix for Fig. 5 according to the different criteria.

6. Method

6.1. Objectives

The aim of this study is to determine which IRT model better fits in the ITS case and the best criterion to construct the performance matrix. We explore different aspects of the calibration process with the input data from learning environments. The challenge is to analyze the best strategy to optimize the calibration results according to the aspect studied. More concretely, we analyze different aspects of calibration, trying to answer the following four questions:

1. *Which IRT model best fits the datasets?* In this sense, we analyze the most extended models for modeling characteristic curves, i.e. 1PL, 2PL and 3PL in order to see which one best fits data for our ITS.
2. *Which is the best strategy to filter raw data for the performance matrix and reduce noise?* In the next section we introduce three filtering criteria for this purpose. We explore which one leads to the best calibration of performance results.
3. *Which is the best strategy for grouping data?* We also explore several strategies for grouping data from different students' samples. In learning environments data collection is usually performed incrementally and this fact needs to be taken into account to guarantee that calibration is accomplished suitably.
4. *When using the CK-session approach, how should be the T_{CK} threshold value?* As explained CK-session criterion can be configured in terms of the threshold considered. In this section we study how the selection of the T_{CK} value influences the calibration performance. That is, our study is focused on analyzing the value for which the T_{CK} can produce a more accurate calibration of constraints.

6.2. Participants

The data considered in this study were obtained from a total of 197 students that used SQL-Tutor as a ITS at the University of Canterbury, New Zealand: 39 students in 2008, 98 in 2009, and 60 in 2010. A first filtering process removed data about 15 students from 2009 and 6 students from 2010 due to their low activity in the system.

Students worked with SQL-Tutor over the course and solved as many problems as they wanted. That generates a huge amount of data in terms of constraints. Different problems were included in each instance of the course, and the set of constraints was modified from 2008 to 2009. Some constraints were the same, some were removed and new ones were added. In this situation, we decided to calibrate the models independently for each year. This decision also allows an analysis of the consistency of the comparison results, which should not differ from one year to the next.

6.3. Procedure

We have assembled the data of each year according to the three filtering scenarios, resulting in 9 initial datasets. Each dataset was extracted from two output files generated by SQL-Tutor with information about the student model. One of those files contained: (1) the list of the problem identifiers solved by the student; (2) for each constraint, the number of times it was relevant; (3) the

Table 1
Number of constraints involved in each scenario.

Filtering scenario	2008	2009	2010
S1	493	502	480
S2	429	386	357
S3	300	478	346

Table 2a
Average negative-twice-the-loglikelihood values by IRT model and year.

IRT model	2008	2009	2010	Average
1PL	4019.04	7852.47	3541.52	5137.68
2PL	3472.58	7094.42	3214.04	4593.68
3PL	3484.96	6993.40	3254.08	4577.48
Average	**3658.86**	**7313.43**	**3336.55**	**4769.61**

Table 2b
Average negative-twice-the-loglikelihood values by IRT model and filtering scenario.

IRT model	S1	S2	S3	Average
1PL	6144.41	6050.18	3218.44	5137.68
2PL	5520.28	5419.03	2841.73	4593.68
3PL	5510.31	5397.14	2824.99	4577.48
Average	**5725.00**	**5622.11**	**2961.72**	**4769.61**

Table 2c
Average negative-twice-the-loglikelihood values by IRT model and grouping strategy.

IRT model	CK (10 min)	CK (5 min)	CK (3 min)	CK (1 min)	1st-time	Problem	Average
1PL	6390.08	6540.91	6693.37	6873.27	3340.36	988.09	5137.68
2PL	6073.51	6177.12	6305.77	6424.66	3279.29	−698.26	4593.68
3PL	6096.12	6210.83	6344.43	6447.49	3243,87	−877.88	4577.48
Average	**6186.57**	**6309.62**	**6447.86**	**6581.80**	**3287,84**	**−196.01**	**4769.61**

list of trials, i.e. the problems in which the constraint was presented; and (4) whether it was violated or not. The other file was a log file that included the problem selected by the student and the date when it was chosen. Also included was the set of constraints that were relevant each time the solution was corrected, the date when it happened and whether or not they were violated. We developed a procedure for combining these two files and generating a dataset containing the set of problem identifiers solved by the student, their relevant constraints, their violations, and their timestamps. The 9 datasets produced 54 different performance matrices applying the 6 different grouping criteria explained in the previous section (4 CK-session criteria with different thresholds, 1st-time-relevant, and problem grouping). Finally, the CCCs calibration was carried out for each of the three different IRT models, i.e., 1PL, 2PL and 3PL. Finally, the computation was performed using Multilog [47]. The whole process of filtering the initial dataset, generating the performance matrices and calibrating them with Multilog, could not be done manually due to the dimension of the data. They were carried out using an auxiliary Java application that performed each step and applied the different factor combinations.

As a result, we obtained a total of $3 \times 3 \times 6 \times 3 = 162$ sets of calibrated CCCs. In order to evaluate the quality of every resulting characteristic curve dataset, we took the negative-twice-the-loglikelihood. This value is twice the log of likelihood function; the lower its value, the better the fit of the dataset [17]. The negative-twice-the-loglikelihood is commonly used as a measure of the goodness of fit for the parameters representing every characteristic curve [11]. It is one of the output values produced by Multilog.

With respect to the CK-session criterion, the data from students who made at least one attempt were used to calibrate the constraints using different values of the threshold, T_{CK}, to generate the virtual students. Precisely, T_{CK} was determined to be 10, 5, 3 and 1 min. The main reason to choose these low values was that learning takes place when the student is solving a problem, and therefore, knowledge does not remain constant for long. It should be noted that the higher the T_{CK}, the lower the number of CK-sessions, and thus, the amount of data for calibration is reduced. On the other hand, if we consider a low T_{CK}, the CK-sessions could be too short, that is, containing only a few constraints and, thus, reducing calibration quality. For this reason, experiments conducted serve to determine the most appropriate value of T_{CK}.

7. Results

7.1. IRT models

In order to determine the model that best fits the data, we have compared the value obtained for the negative-twice-the-loglikelihood in the 162 calibrations. Table 1 shows the number of constraints involved in each filtering scenario for each year.

To compare two values of the negative-twice-the-loglikelihood, we have to find out the degrees of freedom of the χ^2 distribution, which depends on the number of parameters involved in each model (see Section 2.3.). For instance, between 1PL and 2PL, for a given year, the difference is exactly the same as the number of constraints, because each 2PL curve has and additional parameter. The number of constrains varies from one year to another and depends on the filtering scenario, (see Table 1), so for example, for the year 2008, and scenario 3, we should consider χ^2 with 300 degrees of freedom ($\chi^2(300) = 325.40$ for $p = 0.05$) and compare the negative-twice-the-loglikelihood value obtained in the calibration of the 1PL model for that year using a given grouping strategy with the equivalent data obtained from calibration of 2PL model. On the other extreme, considering filtering scenario 1 in the year 2009, will lead to 502 degrees of freedom, which means ($\chi^2(500) = 553.13$ for $p = 0.05$). As an approximation we can say that in the average case a difference in the negative-twice-the-loglikelihood values greater than 400 will indicate a statistically significant difference ($p < 0.05$).

Tables 2a–2c show these values across different combination of the other conditions. Each value in Tables 2a and 2b is the average of the value obtained in 18 calibrations. Each value in Table 2c is the average of 9 calibrations.

According to this reasoning, we can conclude that 1PL produced a calibration with significantly lower quality than the other two regardless of the other conditions. Nevertheless, we could not find any significant difference between the 2PL and 3PL models. Moreover, following a random pattern, sometimes 3PL was better and, at other times, 2PL was better. This suggests that for calibration of constraints, a 3PL model or 2PL perform similarly, but 1PL is not suitable. This is not a surprise, since the difference between 2PL and 3PL is just the guessing factor. Guessing factor applies when the student can solve an item just by random selection, such as a multiple choice item, but it makes no sense talking about guessing for a constraint, since possible outcomes (satisfaction or violation) cannot be randomly selected. As a result, the 2PL model is chosen.

7.2. Filtering scenarios

Tables 3a and 3b contain the average of the negative-twice-the-loglikelihood values by year and grouping method respectively. In this case only data from 2PL RT model have been used, because these have been found to be the most accurate in the previous

Table 3a
Average negative-twice-the-loglikelihood values by filtering scenario and year.

Scenario	2008	2009	2010	Average
S1	5078.47	7850.67	3631.72	5520.28
S2	5032.78	7725.33	3498.97	5419.03
S3	306.48	5707.27	2511.45	2841.73
Average	**3472.58**	**7094.42**	**3214.04**	**4593.68**

Table 3b
Average negative-twice-the-loglikelihood values by filtering scenario and grouping strategy.

Scenario	CK (10 min)	CK (5 min)	CK (3 min)	CK (1 min)	1st-time	Problem	Average
S1	6668.93	6844.77	7010.47	7288.60	3420.03	1888.90	5520.28
S2	6582.07	6713.60	6884.90	7160.73	3420.33	1752.53	5419.03
S3	4969.53	4973.00	5021.93	4824.63	2997.50	−5736.20	2841.73
Average	**6073.51**	**6177.12**	**6305.77**	**6424.66**	**3279.29**	**−698.26**	**4593.68**

Table 4
Number of virtual students involved in each grouping strategy.

Grouping strategy	2008	2009	2010
CK (10 min)	167	293	171
CK (5 min)	190	308	175
CK (3 min)	215	323	178
CK (1 min)	338	370	192
1st time	39	83	54
Problem	1542	1976	938

subsection. Each value in these tables is the average of 6 and 3 values respectively.

Comparing the different scenarios, the latter gives a better result, which suggests that filtering some constraints that are relevant very often is also a good criterion. This issue is especially apparent in the 2008 dataset, where some of the constraints were always relevant, which made them unsuitable for calibration (some positive or negative evidence should occur to produce a suitable calibration result). The filtering of those constraints drastically improved the quality of the calibration.

Note that to compare two filtering scenarios the degrees of freedom of the χ^2 distribution should be determined depending on the model, for the 2PL model the different number of constraints between filtering scenarios S2 and S3 leads to a $2 \times 129 = 258$ additional parameters $(\chi^2(250) = 287.88$ for $p = 0.05)$. The results indicate that the conclusions are very significant with $p << 0.05$

Moreover, the quality of the resulting datasets in each filtering scenario could be related to the number of constraints involved in it. Following this hypothesis, the quality of the constraints should be higher in larger datasets since the fitting error would be lower due to a larger number of evidence. Nevertheless, as we can see in Table 1, that is not true: filtering scenario 3 has fewer constraints than filtering scenario 2 but the quality is higher (see Table 3a), which suggests that the filtering criteria actually remove from the study those constraints that do not provide important information.

7.3. Grouping strategy

According to the grouping strategy the number of students varies from one condition to another (see Table 4). More students implies more parameters (1 by each student), and the higher the number of parameters, the lower expected negative-twice-log-likelihood. Once again, we use the χ^2 distribution to determine if these differences are significant.

Table 5
Average negative-twice-the-loglikelihood values by grouping strategy and year.

Grouping strategy	2008	2009	2010	Average
CK (10 min)	3278.80	8168.30	3461.50	4969.53
CK (5 min)	3217.80	8242.00	3459.20	4973.00
CK (3 min)	3224.00	8335.50	3506.30	5021.93
CK (1 min)	2452.10	8394.40	3627.40	4824.63
1st time	1997.60	4591.40	2403.50	2997.50
Problem	−12331.40	−3488.00	−1389.20	−5736.20
Average	**306.48**	**5707.27**	**2511.45**	**2841.73**

Table 5 shows the average negative-twice-the-loglikelihood values for the 18 calibrations that used the 2PL IRT model applying conditions of filtering scenario S3. The first four rows correspond to the CK-session grouping strategy with different T_{CK} threshold values, while the other two corresponds to the “first time relevant” method and the grouping by problems criteria, respectively.

With respect to the best way to construct the performance matrix, the “first time relevant” grouping criterion performs better than any other CK-session strategy, irrespective of the T_{CK} values. The results are very significant with $p << 0.05$. In general, the lower the T_{CK} value, the better the calibration quality, but these results are not always significant.

However, the grouping by problems criterion outperforms the “first time relevant” grouping criterion. Even considering the higher degrees of freedom for the χ^2 distribution, $(\chi^2(1000) = 1074.68$ for $p = 0.05)$, the results indicate that this strategy leads to statistically significant IRT model fit.

These results could be explained by the fact that the method of grouping constraints by problems produces a larger number of virtual students. This implies that our CK-session method is not appropriate for calibration independently of the T_{CK} values. Instead, the original approach of “the first time relevant” is a better option. The idea of the CK-session is a too coarse-grained methodology to be used in the calibration process of CBM+IRT and, thus, a more fine-grained one, such as the grouping of evidence by problems produces better-quality calibration.

8. Conclusions

Assessment is an important part of any learning process since it is used as a way to determine the starting knowledge state of the student, how this knowledge evolves during the instruction and, at the end of this process, to compute the level of achievement. In computer-based educational research, one of the challenges is the construction of problem-based environments. Automatic assessment of these kinds of tasks (i.e. the problems or complex exercises) is complicated due to the complexity of the knowledge required to be applied by the student. The combination of CBM and IRT can be used as a well-founded approach for this type of assessment.

When the technique is applied to the data of a CBM system for learning purposes with a large number of students using the system and multiple sessions over long periods of time, some limitations have to be taken into account. The main limitation is related to the way in which characteristic curves are calibrated. Calibration is an important previous stage when assessment is accomplished with data-driven theories such as IRT. One of the requirements of IRT to accomplish calibration is to have available datasets of students' performance where the knowledge of each individual had to be kept constant. This means that during the process of collecting this information, no learning could happen. This requirement is difficult to satisfy when data is taken from learning environments.

To study the applicability of different calibration strategies in a real environment, we used log data from SQL-Tutor collected over three years. To guarantee that this principle is met we have introduced two concepts, i.e. the “CK-session” and the “virtual student”, and described three grouping strategies to construct performance matrices from the raw data obtained from the ITS to be used to calibrate the IRT models. Additionally, some data filtering was needed to reduce the “noise” of the data obtained from a ITS. The main conclusion is that better results are obtained by discarding constraints with low variability, and that the IRT models are better adjusted if we consider a “virtual student” for each resolution of a single problem in the ITS. Gathering evidence through problems would produce higher-quality CCCs during the calibration phase.

In addition, we have explored the performance of the three most commonly used IRT models. The goodness of model fit has been measured using the output of the Multilog tool with different combinations of assembling criteria. The results suggest that the 2PL model is the most suitable to for use with CBM constraints in all cases, and that there is no reason to use the 3PL model, which requires more data to be calibrated and fails to provide any significant improvements.

In order to implement any of these calibration approaches in future ITS the conclusions obtained in the study presented here could be taken into account as a guideline. The utilization of these techniques produces a more accurate calibration of the basic elements of the system knowledge base, the CCCs. Furthermore, we would like to explore the performance of this methodology in an ITS to study the improvement in terms of learning that this approach could provide.

Acknowledgments

This work has been partially funded by the Consejería de Economía, Innovación y Ciencia of the Junta de Andalucía through the research projects DEDALO (P09-TIC-5105) and TIADA (P07-TIC-03243).

References

[1] R.G. Almond, R.J. Mislevy, L.S. Steinberg, D. Yan, D.M. Williamson, An introduction to evidence-centered design, Bayesian Networks in Educational Assessment, Springer, New York, 2015, pp. 19–40.
[2] J.R. Anderson, C.J. Lebiere, The Atomic Components of Thought, Psychology Press, 1998.
[3] E.L. Baker, R.E. Mayer, Computer-based assessment of problem solving, Comput. Hum. Behav. 15 (3) (1999) 269–282.
[4] R.S. Baker, A.B. Goldstein, N.T. Heffernan, Detecting the moment of learning, in: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, Saarbrücken, Germany, 2010, pp. 25–34.
[5] J.T. Behrens, R.J. Mislevy, K.E. DiCerbo, R. Levy, An evidence centered design for learning and assessment in the digital world, in: Technology-Based Assessments for 21st Century Skills: Theoretical and Practical Implications From Modern Research, Information Age Publishing Inc., 2012, pp. 13–53.
[6] J. Biggs, Teaching for Quality Learning at University, 2nd ed., SRHE/Open University Press, Buckingham, 2003.
[7] A. Bunt, C. Conati, Probabilistic student modelling to improve exploratory behaviour, User Model. User-Adapt. Interact. 13 (3) (2003) 269–309.
[8] K. Chrysafiadi, M. Virvou, Student modeling approaches: a literature review for the last decade, Expert Syst. Appl. 40 (11) (2013) 4715–4729.
[9] C. Conati, A. Gertner, K. Vanlehn, Using Bayesian networks to manage uncertainty in student modeling, User Model. User-Adapt. Interact. 12 (4) (2002) 371–417.
[10] A.T. Corbett, J.R. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, User Model. User-Adapt. Interact. 4 (4) (1994) 253–278.
[11] R.J. de Ayala, Theory and Practice of Item Response Theory, The Gilford Press, 2009.
[12] A.A. Davier, P.F. Halpin, Collaborative problem solving and the assessment of cognitive skills: psychometric considerations, ETS Res. Rep. Ser. 2013 (2) (2013) i–36.
[13] J. Gálvez, E. Guzmán, R. Conejo, A blended E-learning experience in a course of object oriented programming fundamentals, Knowledge-Based Syst. 22 (4) (2009) 279–286.
[14] J. Gálvez, E. Guzmán, R. Conejo, Data-driven student knowledge assessment through ill-defined procedural tasks, in: Proceedings of International Conference on Current Topics in Artificial Intelligence, CAEPIA 2009, vol. 5988, 2009, pp. 233–241.
[15] J. Gálvez, E. Guzmán, R. Conejo, E. Millán, Student knowledge diagnosis using item response theory and constraint-based modeling, in: Proceedings of *14th* International Conference *on Artificial Intelligence in Education*, vol. 200, 2009, pp. 291–298.
[16] R.K. Hambleton, R.W. Jones, Comparison of classical test theory and item response theory and their applications to test development, Educ. Measur. Issues Pract. 12 (3) (1993) 38–47.
[17] R.K. Hambleton, H. Swaminathan, H.J. Rogers, Fundamentals of Item Response Theory, Sage Publications, Inc., Thousand Oaks, 1991.
[18] D.H. Jonassen, Instructional design models for well-structured and III-structured problem-solving learning outcomes, Educ. Technol. Res. Dev. 45 (1) (1997) 65–94.
[19] Livingston, S.A. (2009). Constructed-Response Test Questions: Why We Use Them; How We Score Them. R&D Connections. Report number 11. Educational Testing Service.
[20] M. Mathews, A. Mitrovic, D. Thomson, Analysing high-level help-seeking behaviour in ITSs, in: Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 2008, pp. 312–315.
[21] R.E. Mayer, Problem solving, in: M.W. Eysenck (Ed.), The Blackwell Dictionary of Cognitive Psychology, Basil Blackwell, Oxford, England, 1990, pp. 284–288.
[22] R.E. Mayer, Thinking, Problem Solving, Cognition, second edition, Freeman, New York, 1992.
[23] R.E. Mayer, M.C. Wittrock, Problem-solving transfer, in: D.C. Berliner, R.C. Calfee (Eds.), Handbook of educational psychology, New York, Macmillan, 1996, pp. 47–62.
[24] M. Mayo, A. Mitrovic, Optimising ITS behaviour with Bayesian networks and decision theory, Int. J. Artif. Intell. Educ. 12 (2001) 124–153.
[25] E. Millán, J.L. Pérez-de-la-Cruz, A Bayesian diagnostic algorithm for student modeling, User Model. User-Adapt. Interact. 12 (2/3) (2002) 281–330.
[26] E. Millán, T. Loboda, J.L. Pérez-de-la-Cruz, Bayesian networks for student model engineering, Comput. Educ. 55 (4) (2010) 1663–1683.
[27] R.J. Mislevy, G.D. Haertel, Implications of evidence-centered design for educational testing, Educ. Measur.: Issues Pract. 25 (4) (2006) 6–20.
[28] R.J. Mislevy, L.S. Steinberg, R.G. Almond, On the structure of educational assessments, Measur.: Inter-discip. Res. Perspect. 1 (2003) 3–67.
[29] R.J. Mislevy, R.G. Almond, J.F. Lukas, A Brief Introduction to Evidence Centered Design. CSE Report 632, The National Center for Research on Evaluation, Standards, and Student Testing, University of California, Los Angeles, 2003.
[30] R.J. Mislevy, M.M. Riconscente, D.W. Rutstein, Design Patterns for Assessing Model-Based Reasoning (PADI-Large Systems Technical Report 6), SRI International, Menlo Park, CA, 2009.
[31] A. Mitrovic, Experiences in implementing constraint-based modeling in SQL-tutor, Intell. Tutoring Syst. Conf. Lect. Notes Comput. Sci. 1452 (1998) 414–423.
[32] A. Mitrovic, Large-scale deployment of three intelligent web-based database tutors, J. Comput. Inf. Technol. 14 (4) (2006) 275–281.
[33] A. Mitrovic, Fifteen years of constraint-based tutors: what we have achieved and where we are going, User Model. User-Adapt. Interact. 22 (1–2) (2012) 39–72.
[34] A. Mitrovic, S. Ohlsson, Implementing CBM: SQL-Tutor after fifteen years, Int. J. Artif. Intell. Educ. 25 (4) (2015) 1–10.
[35] A. Mitrovic, K.R. Koedinger, B. Martin, A comparative analysis of cognitive tutoring and constraint-based modeling, in: User Modeling 2003, Springer, Berlin, Heidelberg, 2003, pp. 313–322.
[36] A. Mitrovic, A. Weerasinghe, Revisiting Ill-definedness and the consequences for ITSs, in: Proceedings of the 14th International Conference on Artificial Intelligence in Education, 2009, pp. 375–382.
[37] A. Mitrovic, A. Mayo, P. Suraweera, B. Martin, Constraint-based tutors: a success story, in: Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 2001, pp. 931–940.
[38] A. Mitrovic, B. Martin, P. Suraweera, Intelligent tutors for all: the constraint-based approach, IEEE Intell. Syst. 22 (2007) 38–45.
[39] S. Ohlsson, Constraint-based student modeling, in: Student Modeling: The Key to Individualized Knowledge-based Instruction, Springer-Verlag, 1994, pp. 167–189.
[40] S. Ohlsson, Learning from performance errors, Psychol. Rev. 103 (2) (1996) 241–262.
[41] P. Reimann, M. Kickmeier-Rust, D. Albert, Problem solving learning environments and assessment: a knowledge space theory approach, Comput. Educ. 64 (2013) 183–193.
[42] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, third edition, Pearson, 2009.
[43] J. Self, The defining characteristics of intelligent tutoring systems research: ITSs care, precisely, Int. J. Artif. Intell. Educ. 10 (1999) 350–364.
[44] J.A. Shapiro, An algebra subsystem for diagnosing students' input in a physics tutorin system, Int. J. Artif. Intell. Educ. 15 (3) (2005) 205–228.
[45] J. Sweller, J. Van Merrienboer, J. Paas, Cognitive architecture and instructional design, Educ. Psychol. Rev. 10 (3) (1998) 251–296.
[46] C.Y. Ting, S. Phon-Amnuaisuk, Properties of Bayesian student model for INQPRO, Appl. Intell. 36 (2) (2012) 391–406.
[47] D. Thissen, MULTILOG User's Guide: Multiple, Categorical Item Analysis And Test Scoring Using Item Response Theory, Scientific Software International, 1991.

[48] D. Thissen, W-H. Chen, R.D. Bock, Multilog (version 7), Scientific Software International, Lincolnwood, IL, 2003.

[49] D.J. Weiss (Ed.), New Horizon Testing: Latent Trait Test Theory and Computerized Adaptive Testing, Elsevier, 2014.

[50] M. Xenos, Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks, Comput. Educ. 43 (4) (2004) 345–359.

[51] J.D. Zapata-Rivera, Indirectly visible Bayesian student models, in: Proceeding of Bayesian Modeling Applications Workshop, BMA, 2007.

[52] X. Zhong, H. Fu, H. Xia, L. Yang, M. Shang, A hybrid cognitive assessment based on ontology knowledge map and skills, Knowledge-Based Syst. 73 (2015) 52–60.