LMS Assessment: using IRT analysis to detect defective multiple-choice test items

Panagiotis Fotaris*
School of Arts and Digital Industries,
University of East London,
Docklands Campus, University Way,
London E16 2RD, UK
Email: p.fotaris@uel.ac.uk
* Corresponding author

Theodoros Mastoras
Department of Applied Informatics,
University of Macedonia,
156 Egnatia str., 54006 Thessaloniki, Greece
Email: mastoras@uom.gr

Abstract: Due to the computerization of assessment tests, the use of Item Response Theory (IRT) has become commonplace for educational assessment development, evaluation, and refinement. When used appropriately by a Learning Management System (LMS), IRT can improve the assessment quality, increase the efficiency of the testing process, and provide in-depth descriptions of item properties. This paper introduces a methodological and architectural framework which embeds an IRT analysis tool in an LMS so as to extend its functionality with assessment optimisation support. By applying a set of validity rules to the statistical indices produced by the IRT analysis, the enhanced LMS is able to detect several defective items from an item pool which are then reported for reviewing of their content. Assessment refinement is achieved by repeatedly employing this process until all flawed items are eliminated.

Keywords: e-learning; item pool optimisation; item response theory; IRT; computer aided assessment; Learning Management System; technology enhanced learning; Massive Open Online Courses; MOOCs.

Biographical notes: Panagiotis Fotaris, PhD, is a Lecturer and the Programme Leader of the BSc (Hons) in Digital Media Design at the School of Arts and Digital Industries, University of East London. He completed his doctorate in Applied Informatics with an emphasis in Technology Enhanced Learning at University of Macedonia and also holds an MA in Graphic Design / Multimedia and an MSc in Distributed and Multimedia Information Systems. His teaching, research, training and consulting interests include gamification, learning analytics, UX design, and the integration of novel technology into the teaching and learning environment.

Theodoros Mastoras, PhD, holds a Diploma in Computer Engineering and Informatics from the University of Patras and a Doctor's degree in Applied Informatics from the University of Macedonia, Greece. His research interests include the areas of data analysis, learning analytics, gamification, e-learning standards, and semantic web technologies. He has given presentations in various international conferences and has published research papers at international journals and conference proceedings.

1 Introduction

Due to the widespread use of software applications and web-based technologies for the administration, documentation, tracking, reporting, and delivery of e-learning education courses (Learning Management Systems – LMS) (Ellis, 2009), as well as the booming development of Massive Open Online Courses (MOOCs), the use of Computer Aided Assessment (CAA) tools has become a major trend in academic institutions worldwide (Anantatmula and Stankosky, 2008; Rego et al., 2009; Virtanen, 2009). Through these systems, tests composed of various question types can be presented to students in order to assess their knowledge (Hindi et al., 2008). However, there has been considerable criticism of the test quality, with both research and experience showing that many test items are flawed at the initial stage of their development due to deviation from widely accepted item-writing guidelines, such as putting the central idea of the question into the stem and avoiding the use of negation whenever possible (Haladyna et al., 2002), long/complex sentences, ineffective distractors (incorrect answers), items with poor discriminatory power or high/low difficulty level etc. Test developers can expect about 50% of the items in their item pool to fail to perform as intended, which may eventually lead to unreliable results of examinee performance (Haladyna, 1999). Thus a critical challenge lies in how to enhance an LMS with a tool which investigates the statistical properties of individual test items and ensures that they are of the highest quality possible, since inferior items yield scores of questionable value that are inappropriate to use as a basis of evaluating student achievement and could therefore threaten the overall effectiveness of the test.

There are two major approaches to item evaluation using item response data and, sample size permitting, both can be used. The first approach uses *Item Analysis* (*IA*) (Hambleton, 1994; Yu and Wong, 2003); it focuses on traditional item indices appearing in *Classical Test Theory* (*CTT*) (SCOREPAK, 2005), which include item difficulty, item discrimination (item effectiveness), and the distribution of examinee responses across the alternative responses. The second approach uses *Item Response Theory* (*IRT*) (Lord, 1980), a framework originally developed to overcome the limitations of CTT, in order to estimate the parameters of an *item-characteristic curve (ICC)* which maps the probability that an item will be answered correctly based on the examinee's ability level as measured by the test.

The natural scale for item difficulty in IA is the percentage of examinees correctly answering the item. One descriptor of item difficulty is *p-value*, which stands for the proportion of the percentage of examinees correctly answering the item. Every item has a natural difficulty based on the performance of all individuals undertaking the test; however, this *p*-value is quite difficult to estimate accurately unless a highly representative group of test-takers is being tested. If, for example, the sample contains well-instructed, highly able or highly trained individuals, then the test and its items will appear very easy. Alternatively, if the sample contains uninstructed, low-ability or untrained individuals, then the same test will appear very hard. Therefore, the p-value is not an invariant characteristic of the item, but it is potentially biased by the sample on which the estimate of item difficulty is based. As a result, the characterization of an item or test is examinee (sample) dependent (Hambleton et al., 1991), while with IRT the composition of the sample is generally immaterial, and item difficulty can be estimated without bias, which can be quite useful when reusing a test a number of times. From the ICC it is clear how the items work and which ability an examinee has that performs well on each item.

In comparison to IRT, IA is also not as sensitive to items that discriminate differentially across different levels of ability, does not work as well when different examinees take different sets of items, and is not as effective in identifying items that are statistically biased (Hambleton & Jones, 1993; Hambleton & Swaminathan, 1987; Schmeiser & Welch, 2006). Hence, the use of IRT when designing tests in an LMS is likely to produce more reliable results.

Although within IRT there are numerous models, including uni- and multi- dimensional models as well as a mixture of distribution models, three prominent equations termed *1PL*, *2PL*, and *3PL* (*parameter logistic*) models are presently used to make predictions. These use one parameter

(*theta* – θ) to measure how much of a latent trait an examinee has (i.e., the amount of ability, trait, proficiency or attribute level possessed by an individual). In 3PL, each item is characterized by the three parameters, α, *b* and *c* respectively.

In a cognitive task, the α parameter indicates the degree to which an examinees' response to an item varies with, or relates to their trait level or ability (Nenty, 2004). It is a measure of the discriminating power of the item. Although it is defined theoretically on the scale (-∞, +∞) with the usual range seen in practice being -2.80 to 2.80 (Baker, 2001), negatively discriminating items are discarded from ability tests. If for example, the probability of answering an item correctly decreases as examinee ability increases, something is wrong with that particular item (such as mis-keying).

The *b* parameter is the amount of trait inherent in an item and represents the cognitive resistance of the item or task. It serves as an index of item difficulty and increases in value as items become more difficult. The theoretical range of values is (-∞, +∞), however typical values are ranged between [-3, 3] (Baker, 2001). In contrast to the *p*-value used in IA, *b* is theoretically not dependent on the ability level of the sample of students tested.

Finally, the *c* parameter is commonly called the *guessing* or the *pseudo-guessing* parameter and characterises the lower asymptote at which a person completely lacking in the trait will overcome or answer the item correctly. The latter term is used in order to emphasize that guessing, in particular random guessing on selected-response (e.g. multiple-choice) test items, may not be the psychological mechanism by which very low ability examinees are producing correct answers. For example, for a 4-choice item, random guessing would produce a probability of producing a correct response of 0.25. However, it is not uncommon for *c* to assume values smaller than 0.25 because examinees with partial knowledge are attracted by well-constructed distractors that reflect their misunderstandings (Lord, 1974). The *c* parameter has a theoretical range of [0, 1], but in practice values above 0.35 are not considered acceptable (Baker, 2001).

All three parameters are present in the following equation called *Item Response Function (IRF)* that defines the 3PL model for dichotomous data. IRF gives the probability of a correct response to item *i* by an examinee with ability θ:

$$P_i(\theta) \equiv P_i(X_i = 1|\theta) = c_i + \frac{(1-c_i)}{1+e^{-D\alpha_i(\theta-b_i)}} \quad i = 1, 2, \ldots, n \tag{1}$$

In Equation (1), X_i is the score for item i, with $X_i = 1$ for a correct response and $X_i = 0$ for an incorrect response. θ is the examinee's *proficiency*, α_i, b_i, and c_i are item parameters, and D is a scaling constant, In the case of a typical test item, this probability will be small for examinees of low ability and large for examinees of high ability. If one plotted $P(\theta)$ as a function of ability, the result would be the Item Characteristic Curve, a smooth S-shaped curve which describes the relationship between the probability of a correct response to an item and the ability scale (Baker, 2001). The difficulty of an item describes where the item functions along the ability scale, e.g., an easy item functions among the low-ability examinees and a hard item among the high-ability examinees, respectively. The item's discrimination is proportional to the slope of the ICC at $\theta = b$, and the lower limit of the ICC is the value of the guessing parameter c (Fig. 1).

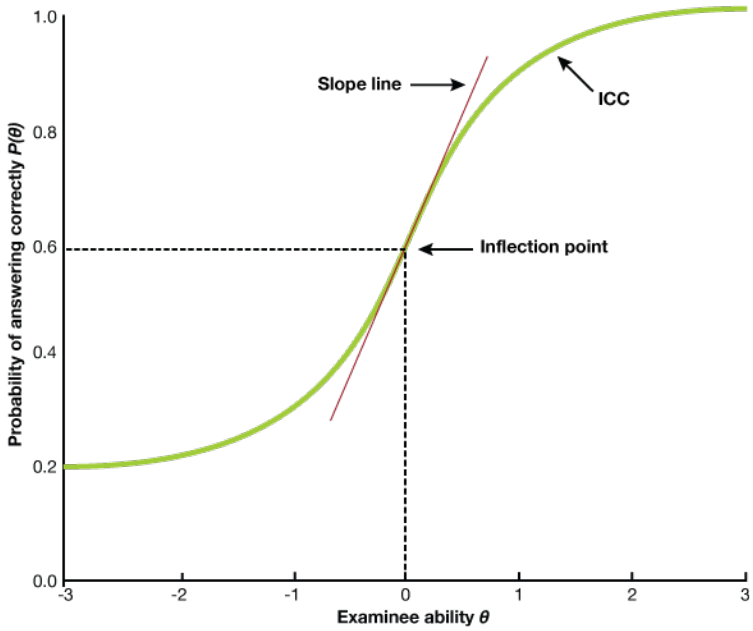

Fig. 1: 3PL Item Response Function ($\alpha = 1$, $b = 0$, $c = 0.2$)

A satisfactory pool of items for testing is one characterized by items with high discrimination ($\alpha > 1$), a rectangular distribution of difficulty (b), and low guessing ($c < 0.2$) parameters (Baker, 1992; Flaugher, 2000). The information provided by the item analysis assists not only in evaluating performance but in improving item quality as well. Test developers can use these results to discriminate whether an item can be

reused as is, should be revised before reuse, or should be taken out of the active item pool. What makes an item's performance acceptable should be defined in the test specifications within the context of the test purpose and use.

The present paper introduces a comprehensible way to present IRT analysis results to test developers without delving into unnecessary details. Instead of executing complex commands and memorising scenarios from technical manuals in an effort to construct a test with high-quality items, test developers can easily detect problematic multiple choice questions from the familiar user interface of an LMS.

The latter can automatically calculate the limits and rules for the α (discrimination), b (difficulty), and c (guessing) parameters (Lord, 1980) based on the percentage of questions wanted for revision. Regarding the examinee's proficiency (θ), while it can be measured on a scale having a midpoint of zero, a unit measurement of one, and a range from negative infinity to positive infinity, practical considerations usually limit the range of values from -3 to +3 (Baker, 2001). However, since these scores include negative ability estimates which will undoubtedly confuse many users, they can optionally be normalized to a 0…100 range scale score.

2 Related Work

Students' increasing demand for more flexible learning options during the last decade has led to the widespread use of LMS and CAA tools in education, and, more recently, to the rapid expansion of MOOCs distributed in platforms such as Coursera, Udacity, FutureLearn, and EdX. However, there is serious concern around the assessment of student learning due to the fact that only a small fraction of the aforementioned systems supports an assessment quality control process based on the interpretation of item statistic parameters. Popular e-learning platforms such as Moodle and Blackboard have plug-ins or separate modules that provide statistics for test items, but apart from that they offer no suggestions to test developers on how to improve their item pool. Similarly, although new web technologies allow for scalable ways to deliver video lectures, implement social fora, and track student progress in MOOCs (Piech et al., 2013), there is limited feedback regarding the quality of the test items and the accuracy of the assessment results. Therefore, many researchers have recently endeavoured to provide mechanisms for assessment optimisation.

Hsieh et al. (2003) introduced a model that presents test statistics and collects students' learning behaviours for generating analysis result and

feedback to tutors. Hung et al. (2004) proposed an analysis model based on Item Analysis (IA) that collects information such as item difficulty and discrimination indices, questionnaire and question style, etc. These data are combined with a set of rules in order to detect defective items, which are signalled using traffic lights. Costagliola et al.'s eWorkbook system (2008) improved this approach by using fuzzy rules to measure item quality, detect anomalies on the items, and suggest improvements. Nevertheless, all of the aforementioned works preferred IA to IRT due to its ease of use without taking into consideration its numerous deficiencies.

On the other hand, IRT has been mainly applied in the Computerized Adaptive Test (CAT) domain for personalized test construction based on individual ability (Chen et al., 2004; Ho and Yen, 2005; Yen and Fitzpatrick, 2006; Meyer and Zhu, 2013). Despite its high degree of support among theoreticians and some practitioners, IRT's complexity and dependence on unidimensional test data and large samples often relegate its application to experimental purposes only. While a literature review can reveal many different IRT estimation algorithms, they all involve heavy mathematics and are unsuitable for implementation in a scripting language designed for web development (e.g., PHP). As a result, their integration in internet applications such as LMSs is very limited. A way to address this issue is to have a web page call the open-source analysis tool ICL (Hanson, 2002) to carry out the estimation process and then import its results for display. The present paper showcases in detail a framework proposed by Fotaris & Mastoras (2013) that follows this exact method in order to extend an LMS with IRT analysis services at no additional programming cost.

3 Open-source IRT Analysis Tool ICL

Several computer programs that provide estimates of IRT parameters are currently available for a variety of computer environments, including Rascal, Ascal, WINSTEPS, BILOG-MG, MULTILOG, PARSCALE, RUMM and WINMIRA to name a few that are easily obtainable (Meyer and Zhu, 2013). Despite being the de facto standard for dichotomous IRT model estimation, BILOG is a commercial product and limited in other ways. Hanson (2002) provided an alternative stand-alone software for estimating the parameters of IRT models called *IRT Command Language* (*ICL*). A recent comparison between BILOG-MG and ICL (Mead et al., 2007) showed that both programs are equally precise and reliable in their estimations. However, ICL is free, open-source, and licensed in a way that allows it to be modified and extended. In fact, ICL is actually IRT

estimation functions embedded into a fully-featured programming language called *TCL* that supports relatively complex operations. Additionally, ICL's command line nature enables it to run in the background and produce analysis results in the form of text files. Since the proposed framework uses only a binary-scoring 3PL model, ICL proves more than sufficient for our purpose and was therefore selected to complement the LMS for item pool optimisation.

4 Integrating IRT Analysis in Dokeos

Dokeos is an open-source LMS implemented in PHP that requires Apache acting as a web server and MySQL as a Database Management System. It has been serving the needs of two academic courses at the University of Macedonia for over six years, receiving satisfactory feedback from both instructors and students. In order to extend its functionality with IRT analysis and item pool optimisation functions, we had to modify its source code so as to support the following features:

1. After completing a test session, the LMS stores in its database the examinee's response to each test item instead of keeping only a final score by default.
2. Test developers define the acceptable limits for the following IRT analysis parameters: item discrimination (α), item difficulty (b), and guessing (c). To cater for test developers who are unaware of the 3PL model parameters' meaning, the LMS offers online help documentation and guidelines regarding how to set the parameters' valid ranges. It also provides the option of choosing from a small selection of predefined limit values. The LMS stores these values as validity rules for each assessment. There is an additional choice of having these limits set automatically by the system in order to rule out a specific percentage of questions (Fig. 2.1).
3. Every time the LMS is asked to perform an IRT analysis, it displays a page with the estimated difficulty, discrimination and guessing parameters for each test item. If the latter violates any of the validity rules already defined in the assessment profile, it is flagged for review of its content (Fig. 2.2). Once item responses are evaluated, test developers can discard, revise or retain items for future use.
4. In addition to a total score, the assessment report screen displays the proficiency θ per examinee as derived from the IRT analysis (Fig. 2.3).

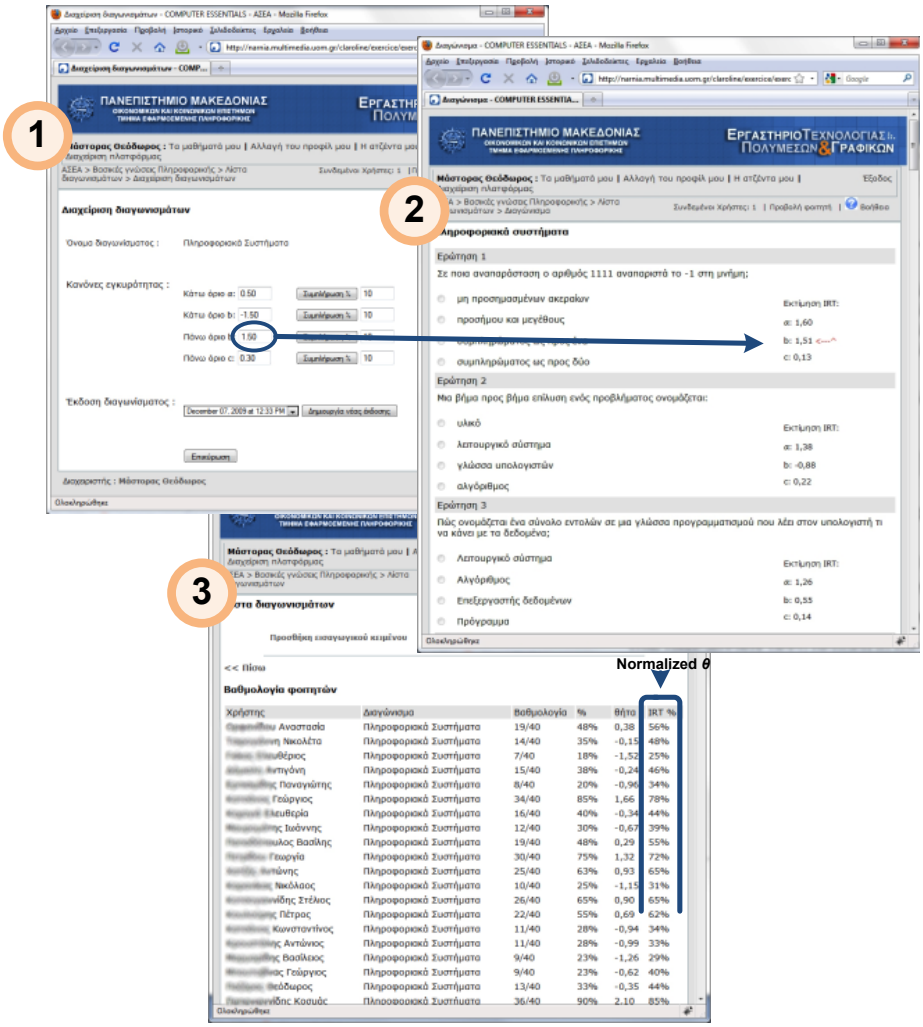

Fig. 2. Functionality features supported in the extended version of Dokeos

The proposed methodology consists of four steps, with each one of them being an action performed by the LMS (Fig. 3). Additionally, the initial database schema has been extended in order to support some extra functions. Once an update of the IRT results is called for, the LMS exports the proper data files and TCL scripts. It then performs a number of calls to the ICL using PHP and after parsing the analysis results, it imports them to its database. A detailed description of the four methodology steps follows:

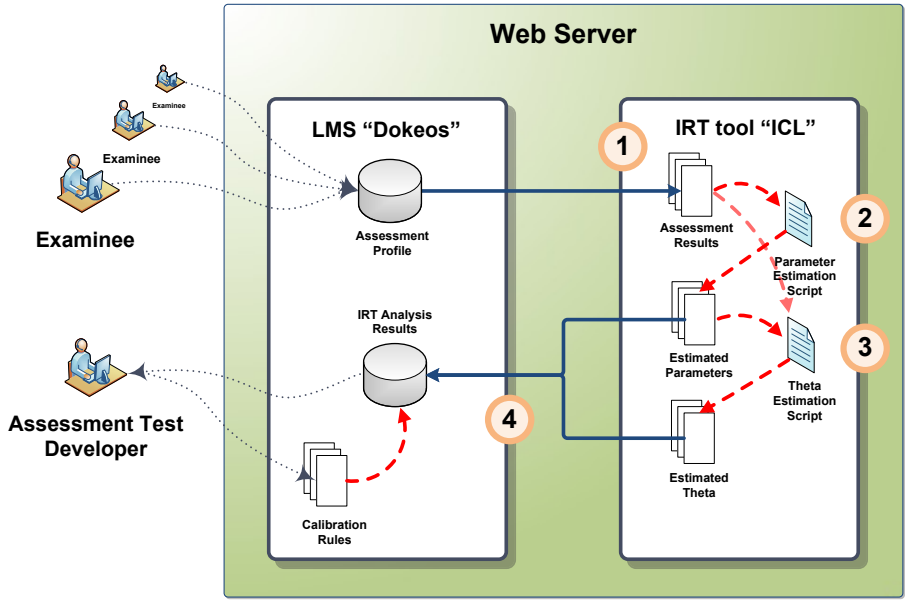

Fig. 3. System architecture

1. The LMS exports the assessment results to a data file and generates a TCL script to process them (parameter estimation script) (Fig. 4).

```
0101001000100111111001010101100000100111
0101000100010001100000111001100000100110
0000000000000011000000110001000010001000
0001010000110010100000111101110010000100
0100010000000001100000000001001010000100
0111011101110111111101111111111101111111
1111001001110000000000011101010000101100
0110000000010011101000110000001000000110
.
.
.
..... one row per examinee .....
```

```
output -no_print
allocate_items_dist 40
read_examinees test0140.dat 40i1
starting_values_dichotomous
EM_steps -max_iter 200
print -item_param
release_items_dist
```

Fig. 4. (a) Assessment results (test0140.dat file). (b) Parameter Estimation Script (test0140.tcl file).

2. The LMS then calls up ICL with the parameter estimation script passed as a parameter in order to create a data file containing the α, b, and c values for each test item. At the same time it prepares a second TCL script to process these IRT parameters (θ estimation script) (Fig. 5).
3. The LMS calls up ICL with the θ estimation script passed as a parameter so as to make a data file with the examinees' θ values (Fig. 6).

```
1  1,597597  1,506728  0,128515
2  1,377810  -0,876164 0,223903
3  1,258461  0,549362  0,140593
4  1,031856  0,495642  0,079279
5  1,077831  1,004437  0,136324
6  0,479151  1,544218  0,218270
7  1,439241  1,279352  0,082382
8  0,898259  1,310215  0,129570
9  1,837514  1,349520  0,032675
10 0,467694  0,934207  0,206085
11 0,607603  0,265524  0,181212
12 0,240009  1,054301  0,245737
13 0,945631  1,451464  0,050895
.
.
.
…… one row per item ……
```

```
output -no_print
allocate_items_dist 40
read_examinees test0140.dat 40i1
read_item_param test0140.par
set estep [new_estep]
estep_compute $estep 1 1
delete_estep $estep
set eapfile [open test0140.theta w]
for {set i 1}{$i <= [num_examinees]}
                        {incr i} {
.
.
.
}
close $eapfile
release_items_dist
```

Fig. 5. (a) Estimated parameters (test0140.par file). (b) θ estimation script (test0140t.tcl file).

```
0,378453      0,434304      19
-0,149162     -0,096175     14
-1,523733     -5,999491     7
-0,238032     -0,172708     15
-0,964941     -1,001566     8
1,658672      1,737581      34
-0,343387     -0,312642     16
-0,665486     -0,666954     12
.
.
.
…… one row per examinee ……
```

Fig. 6. Estimated theta (test0140.theta file)

4. Finally, the LMS imports the two ICL-produced data files (*.par and *.theta) to its database for further processing in the context of the aimed item pool optimisation.

As already mentioned, some modifications to the Dokeos database schema had to be performed in order for the system to function properly. More specifically, while the initial schema supported only a total score per examinee ("*track_e_exercices*" table), the proposed one requires a detailed recording of each examinee's performance per item (Fig. 6). The additional functionalities of this new schema are outlined in the following list:

1. Each assessment can have multiple versions based on its revised items. By monitoring the examinees' performance on each item, test developers can determine whether a certain modification of a specific item affected positively its quality. In practice, each version serves as a new test for the LMS.
2. Each examinee's score per item is recorded for every test being administered. These values are held in the assessment results data file (*.DAT) used by ICL.

3. Test developers can establish a new set of rules for each version of the assessment.

As the main aim of the revised solution is to facilitate further updating processes, the structure and the fields of the initial LMS database have been kept intact, with the only change being the addition of two new tables:

1. Table "*track_e_answers*" stores the examinee's choice per item (fields "*answer_ id*" and "*answer*"), whether this choice was correct (field "*correct*"), and its weight value (field "*weighting*") (Fig. 7.2). Moreover, it supports the recording of multiple responses for future polytomous analyses.

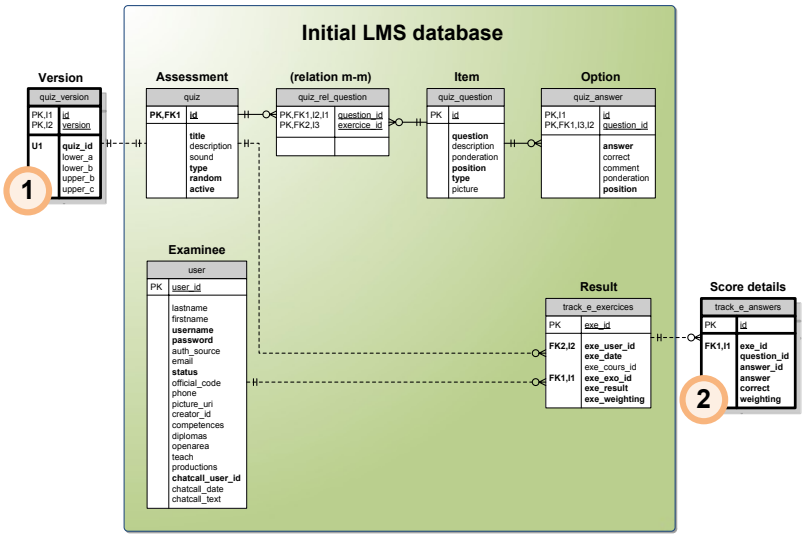

Fig. 7. Entity-Relationship diagram of LMS database extensions

2. Table "*quiz_version*" records each assessment's versions and has a one-to-one relationship to table "quiz" (Fig. 7.1). Additional table entries are added on two occasions:

   (a) When a new assessment is created. In this case the following actions are performed:
      (i) A new record is added to table "*quiz*".
      (ii) A new record is added to table "*quiz_version*". This entry forms the first version of the assessment.
   (b) When a new version of an existing assessment is created. When this occurs, the following course of action is taken:

(i) A new record is added to table “quiz_version”. This entry forms the new version of the assessment.
(ii) A new record is added to table “quiz”.
(iii) All records in table “quiz_rel_question” linking the previous assessment version with its items are copied, so that they remain unaltered in the previous version while being modified in newer versions.
(iv) The entry referring to the previous assessment version in table “quiz” is deactivated; as a result, only the most recent version is available to the examinees. This solution guarantees the preservation of the analysis data related to all previous versions in an easily retrievable format unaffected by subsequent changes.

5 Item Pool Optimisation Process

The proposed system has been implemented by adding the previous features to an existing version of Dokeos at the Department of Applied Informatics, University of Macedonia. A pilot assessment test containing an item pool of 40 questions on “Fundamentals of Information Systems” was developed, including two questions that were purposely flawed (i.e., their difficulty level was too high and too low, respectively) in order to test the system’s detector. Since the test was not connected to an actual university course and contained questions of a general nature, it managed to attract the attention of 113 students who voluntarily participated in the pilot assessment. Before administering the test, the acceptable limits for the IRT parameters were set to $\alpha \geq 0.5$, $-1.7 \leq b \leq 1.7$, and $c \leq 0.25$ respectively (Baker, 2001; Jones & Hurtz, 2004).

The IRT analysis following the completion of the assessment test revealed 9 test items that needed reviewing. In particular, items 6, 10, 12 and 33 showed a low degree of discrimination (Fig. 8), items 21 and 27 appeared too difficult and item 38 deemed too easy (Fig. 9). An extra couple of items (24, 37) were flagged for revision due to their high guessing value (Fig. 10).

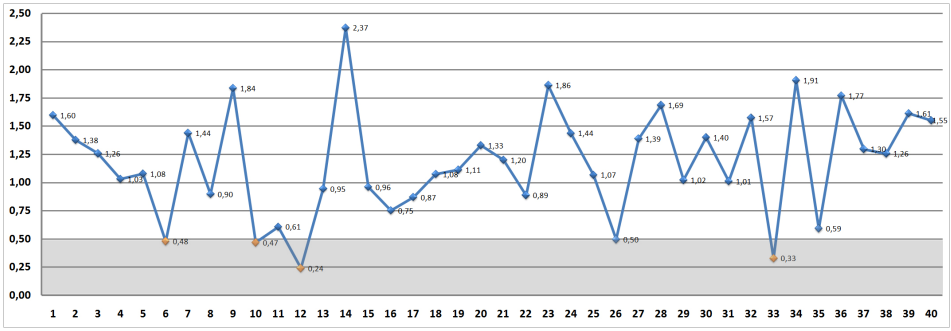

Fig. 8. Item Discrimination Parameter Values (α)

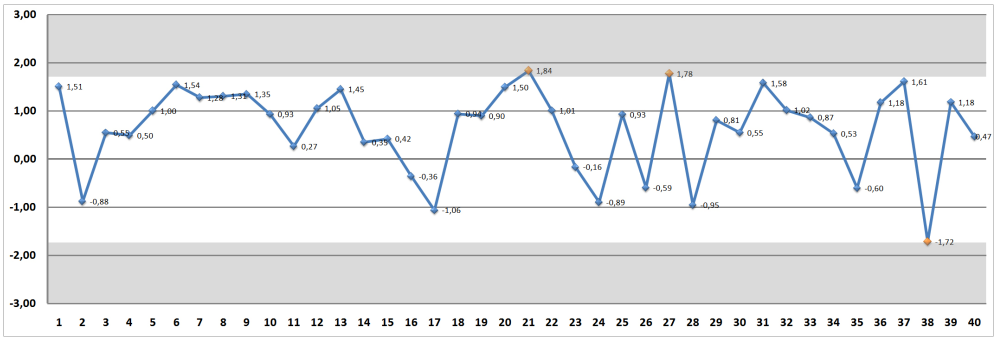

Fig. 9. Item Difficulty Parameter Values (b)

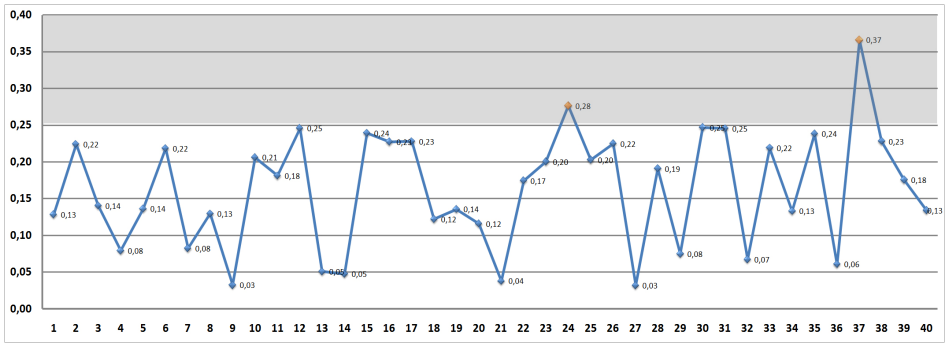

Fig. 10. Item Guessing Parameter Values (c)

Initial Item	Revised Item
Problem: Low level of difficulty; the stem provides a clue to the correct answer.	
Q: In the paged memory allocation scheme, the operating system retrieves data from secondary storage in same-size blocks called:	Q: In which memory allocation scheme does the operating system retrieve data from secondary storage in several blocks of different sizes?
A.pages B. frames C. segments D. partitions	**A.segmented** B. paged C. demand paging D. partitioned
Problem: Low degree of discrimination; the key answer confused examinees of both high and low abilities.	
Q: The transfer layer protocol of TCP/IP is called:	Q: The transfer layer protocol of TCP/IP is called:
A. TCP B. UDP C. IP **D.A and B**	**A.TCP/UDP** B. FTP C. IP D. HTTP
Problem: High guessing value probably due to the graduated answers.	
Q: How many are the basic control structures in programming?	Q: The control structure used to choose among alternative courses of action is called:
A. one B. two C. **three** D. four	A. sequence B. repetition C. **selection** D. iteration

Table 1. Initial and revised versions of defective test items

To elaborate, Table 1 presents 3 test items that were flagged for reviewing based on the IRT analysis results along with their revised versions.

Once an initial item pool has been optimised, examinees can be tested routinely. Such a programme of testing is likely to generate a need to retire flawed, obsolete, or frequently used items, and to replace these with new ones. The extended LMS under consideration detects these problem areas, thus making it easier for test developers to improve the quality of their tests provided that they investigate these issues further and focus on addressing the root cause of the problem in each case (e.g., obscure or ambiguous phrases, obvious correct answer, typographic or logical errors,

a lack of essential information, etc.). In addition, the LMS allows them to create a new version of the assessment test effortlessly by copying the previous iteration and either correcting or replacing whichever items have been flagged as defective. Subsequently, once the revised examination cycle is completed, a new analysis report will ascertain whether all items conform to the validity rules. The number of times a specific assessment must be repeated before leading to a final version with all the problematic items eliminated relies on the comprehension of the analysis results. The faster test developers identify the actual cause of each problem and come up with an appropriate solution, the fewer the necessary iterations.

6 Evaluating the detector

Based on a concept by Baker et al. (2008), we considered a set of four potential criteria in order to evaluate the effectiveness of the proposed system as a detector of defective multiple-choice test items:

First, an ideal detector should accurately identify poorly written multiple-choice test items, e.g., items that contain clues to the correct answer, are too easy, are worded ambiguously, or are purposely flawed so as to test the detector.

Second, provided that flawed items are removed or corrected once identified, successive applications of the detector to the same set of items should yield no more defective items.

Third, the detector should be applicable to any kind of multiple-choice test, regardless of its size or its subject-matter area.

Fourth, after an item is flagged by the detector, it should be relatively easy for the test developer to deduce the item's defect, e.g., great difficulty, low discrimination etc.

The evaluation of the proposed detector was based on how well it addressed the four aforementioned criteria when used on a series of 6 tests. The latter comprised of a set of 28-58 randomly selected items from an item pool of 100. Additionally, each test contained 2 intentionally flawed items: one with completely implausible distractors, and a second with an ambiguous stem. After administering the tests to students, their results were analysed by the detector in order to flag out those items that needed to be substituted. In each case, the IRT analysis correctly identified the intentionally flawed items as being too easy and too hard, respectively (1st criterion). Furthermore, in 3 cases, the detector picked out 4 additional items whose difficulty or discrimination parameter values exceeded the acceptable limits. These items were in turn forwarded to the test developers who were able to identify the root cause of the problem in each

occasion (too plausible distractor, lack of correct answer, ambiguous stem, implausible distractors) (4th criterion). Subsequently, new tests were prepared, containing the same items from the previous tests, except for the discarded ones that were replaced by unused items. These tests were then administered to a new group of students and their results were submitted to the detector for analysis. The same process was carried out for 2 more iterations, when the system eventually ceased to detect new defective items (2nd criterion). Finally, the 3rd criterion is satisfied due to the detector's implementation which allows it to be used for all kinds of tests.

Although there have been several studies on multiple-choice test development practices, the reasons that make test developers produce flawed items is still unclear. With that in mind, the proposed detector can function as a self-reflection tool for teachers that will allow them to improve their skills in constructing well-written multiple-choice items.

7 Conclusion

The present paper introduced a methodological and architectural framework for extending an LMS with IRT–based assessment optimisation. Instead of having web developers implement complex IRT estimation algorithms within the LMS, the proposed methodology uses ICL to obtain reliable IRT analysis results. The latter are then automatically imported into the LMS, thus releasing test developers of this burdensome duty. By applying a set of validity rules, the enhanced LMS acts as a detector that identifies several defective items which are then reported for review of their content. As a result, the suggested approach is capable of assisting test developers in their continuous effort to optimise their item pools. Moreover, the user-friendly interface allows users with no previous expertise in statistics to comprehend and utilise the IRT analysis results.

According to research focused on IRT sample size effects, a great number of examinees are needed to obtain accurate results (Bunderson et al., 1989). For example, Swaminathan and Gifford (1983) concluded that about 1,000 examinees are required when using the 3PL model. Such sample size requirements would normally pose a problem for most test developers due to the fact that the number of examinees in academic courses rarely exceeds 150. However, in cases where instructors are only trying to identify items that are either unrelated to the overall score, too easy, or too difficult, reliable results can be produced even for relatively small classrooms (Fotaris et al., 2011). MOOCs, on the other hand, enrol tens of thousands of students which are more than enough to obtain

accurate estimates with any IRT model. As a result, the proposed system would be ideally suited for a MOOC environment; optimising its extensive item pools will improve the quality of assessment of student learning and could possibly drive more institutions to offer course credit for MOOC completion, thus further expanding the influence of these courses on higher education throughout the world (Meyer and Zhu, 2013).

This initial research project produced encouraging results, showing that the system can effectively evaluate item performance and therefore increase the overall validity of the assessment process. The fact that the proposed methodology is not limited to Dokeos but can be adopted by different e-learning environments (e.g., MOOC platforms) makes it very promising.

References

1. Baker, F.B. (1992) *Item Response Theory: Parameter Estimation Techniques*, Marcel Dekker, New York.
2. Baker, F.B. (2001) *The Basics of Item Response Theory*, 2nd ed., ERIC Clearinghouse on Assessment and Evaluation, University of Maryland, College Park, MD.
3. Baker, R.S.J., Corbett, A.T., Roll, I. and Koedinger, K.R. (2008) 'Developing a generalizable detector of when students game the system', *User Modeling and User-Adapted Interaction*, Vol. 18 No. 3, pp.287-314.
4. Bunderson, C.V., Inouye, D.K. and Olsen, J.B. (1989) 'The Four Generations of Computerized Educational Measurement', in Linn, R.L. (ed.), *Educational Measurement*, Collier Macmillan Publishers, London.
5. Chen, C.M., Duh, L.J. and Liu, C.Y. (2004) 'A Personalized Courseware Recommendation System Based on Fuzzy Item Response Theory', in *IEEE International Conference on e-Technology, e-Commerce and e-Service*, pp.305-308.
6. Costagliola, G., Ferrucci, F. and Fuccella, V. (2008) 'A Web-Based E-Testing System Supporting Test Quality Improvement', paper presented to Advances in Web Based Learning – ICWL 2007.
7. Ellis, R.K. (2009) *Learning Circuits – Field Guide to Learning Management Systems*, American Society for Training and Development (ASTD), Alexandria, VA.
8. Flaugher, R. (2000) 'Item Pools', in H Wainer (ed.), *Computerized Adaptive Testing: A Primer*, 2nd ed., Lawrence Erlbaum Associates, Mahwah, New Jersey.

9. Fotaris, P. and Mastoras, T. (2013) 'Integrating IRT Analysis into LMS for Item Pool Optimization', in *Proceedings of the 2nd Workshop on Technology Enhanced Formative Assessment*, CEUR-WS, Vol. 1147. September 17-21, Paphos, Cyprus.
10. Fotaris, P., Mastoras, T., Mavridis, I. and Manitsaris, A. (2011) 'Identifying Potentially Flawed Items in the Context of Small Sample IRT Analysis', *International Journal On Advances In Intelligent Systems*, Vol. 4 No. 1&2, pp.31-42.
11. Haladyna, T.M. (1999) *Developing and Validating Multiple-Choice Test Items*, 2nd ed., Lawrence Erlbaum Associates, Mahwah, New Jersey.
12. Haladyna, T.M., Downing, S.M. and Rodriguez, M.C. (2002) 'A review of multiple-choice item-writing guidelines for classroom assessment', *Applied Measurement in Education,* Vol. 15, pp.309-344.
13. Hambleton, R.K. (1994) 'Item Response Theory: A Broad Psychometric Framework for Measurement Advances', *Psicothema*, Vol. 6 No. 3, pp.535-556.
14. Hambleton, R.K. and Jones, R.W. (1993) 'Comparison of Classical Test Theory and Item Response Theory and their Applications to Test Development', *Educational Measurement: Issues and Practices*, Vol. 12, pp.38-46.
15. Hambleton, R.K. and Swaminathan, H. (1987) *Item Response Theory: Principles and Applications*, Kluwer-Nijhoff Publishing, Boston.
16. Hanson, B.A. (2002) *IRT Command Language (ICL)*. Obtained through the Internet: http://www.b-a-h.com/software/irt/icl/index.html, [accessed 26/6/2013].
17. Hindi, N.M., Najdawi, M.K. and Jolo, H.A.M. (2008) 'An Examination of Assessment Practices in Colleges of Business at Various Middle East Countries', *International Journal of Teaching and Case Studies*, Vol. 1 No. 4, pp.319-332.
18. Ho, R.G. and Yen, Y.C. (2005) 'Design and Evaluation of an XML-Based Platform-Independent Computerized Adaptive Testing System', *IEEE Transactions on Education*, Vol. 48 No. 2, pp.230-237.
19. Hsieh, C., Shih, T.K., Chang, W. and Ko, W. (2003) 'Feedback and Analysis from Assessment Metadata in E-learning', in *17th International Conference on Advanced Information Networking and Applications (AINA '03)*, pp.155-158.
20. Hung, J.C., Lin, L.J., Chang, W., Shih, T.K., Hsu, H., Chang, H.B., Chang, H.P. and Huang, K. (2004) 'A Cognition Assessment Authoring System for E-Learning', in *24th International Conference on Distributed Computing Systems Workshops (ICDCS 2004 Workshops)*, pp.262-267.

21. Jones, C.N. and Hurtz, G.M. (2004) 'Influences of IRT item attributes on Angoff rater judgments', in 28th Annual IPMAAC Conference on Personnel Assessment, Seattle, WA.
22. Lord, F.M. (1980) *Applications of Item Response Theory to Practical Testing Problems*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.
23. Lord, F.M. (1974) 'Estimation of Latent Ability and Item Parameters when there are Omitted Responses', *Psychometrika*, Vol. 39, pp.247-264.
24. Mead, A.D., Morris, S.B. and Blitz, D.L. (2007) *Open-source IRT: A Comparison of BILOG-MG and ICL Features and Item Parameter Recovery*, Illinois Institute of Technology, Institute of Psychology, Chicago, Unpublished manuscript. Obtained through the Internet: http://mypages.iit.edu/~mead/MeadMorrisBlitz2007.pdf, [accessed 1/7/2013].
25. Meyer, J. P. and 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,* Vol. 8 No. 1, pp.26-39.
26. Nenty, H.J. (2004) 'From Classical Test Theory to Item Response Theory: An introduction to a desirable transition', in Afemikle, O.A., and Adewale, J.G. (eds.), *Issues in Educational Measurement and Evaluation in Nigeria*, Institute of Education, Ibadan, Nigeria.
27. Piech, C., Huang, J., Chen, Z., Do, C., Ng A. and Koller, D. (2013). 'Tuned Models of Peer Assessment', in S. D'Mello, S., Calvo, R. and Olney, A. (eds.), Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013), July 6-9, Memphis, TN, USA.
28. Schmeiser, C.B. and Welch, C.J. (2006) 'Test Development', in Brennan, R.L. (ed.), *Educational Measurement*, 4th ed., Praeger Publishers, Westport, Connecticut.
29. SCOREPAK (2005) 'SCOREPAK: Item Analysis'. Obtained through the Internet: http://www.washington.edu/oea/pdfs/resources/item_analysis.pdf, [accessed 18/1/2014].
30. Swaminathan, H. and Gifford, J.A. (1983) 'Estimation of Parameters in the Three-parameter Latent Trait Model', in Weiss, D.J. (ed.), *New Horizons in Testing*, Academic Press, New York.
31. Virtanen, S. (2009) 'Increasing the self-study effort of higher education engineering students with an online learning platform', *International Journal of Knowledge and Learning*, Vol. 4 No. 6, pp.527-538.

32. Yen, W. and Fitzpatrick, A.R. (2006) 'Item Response Theory', in Brennan, R.L. (ed.), *Educational Measurement (4th edition)*, Praeger Publishers, Westport, Connecticut.
33. Yu, C.H. and Wong, J. W. (2003) 'Using SAS for classical item analysis and option analysis', in *Proceedings of 2003 Western Users of SAS Software Conference*. Obtained through the Internet: http://www.creative-wisdom.com/pub/WUSS2003_classic.pdf, [accessed 18/3/2013].