A Model and Supporting Mechanism for Item Evaluation in Distance Learning-Based Environment

A. M. Ibraheem
Email: to_ahmed54@hotmail.com
National Center of Examinations & Educational Evaluation, El Mokataam, Cairo
K. Shaalan, M. B. Riad, and M. G. Darwish
shaalan@mail.claes.sci.eg, mbriad@cu.edu.eg, gamaldar@frcu.eun.eg
Faculty of Computers and Information, Cairo University

Abstract: Many researchers have observed that 80% -90% of tutorial utterances were in the form of questions, so-called items. However, the item quality in distance learning environment has not been discussed enough. Poor and problematic items may defeat the purpose of distance learning. Therefore, we should ensure the quality and integrity of items before storing them finally in item banks, and becoming available for distance learning systems. The traditional paper-and- pencil process for evaluating an item is performed by administering a pilot test in schools, so-called tryout. Many manual steps are needed to test the items, by trying them, in schools. This process is costly, very time–consuming endeavor, and sometimes inaccurate. In this paper, we attempt to solve this problem by introducing a new practical model for evaluating an item online. Through this model we get, on the spot, student responses and apply some techniques on these responses to identify item characteristics to ensure the quality and integrity of the item. Accordingly, this will enable us to detect and eliminate both weak and problematic items, and store only good items in the item bank, all in quick and accurate manner.

Keywords: distance learning, item banking systems, tutoring systems, item evaluation, online testing.

نموذج وآلية داعمة لتقويم السؤال فى بيئة التعليم عن بعد

لاحظ كثير من الباحثين أن نسبة تتراوح ما بين ٨٠% إلى ٩٠% من الحوارات التى تتم بين الطلاب وأنظمة التعليم عن بعد تكون على شكل أسئلة. ولهذا يجب التأكد من سلامة وجودة تلك الأسئلة قبل تخزينها بشكل نهائى فى بنوك الأسئلة، وتصبح متاحة للطلاب من خلال أنظمة التعليم عن بعد. العملية التقليدية لتقويم السؤال – للتأكد من سلامته وجودته– تتم بإدارة إختبار إستطلاعى فى المدارس على عينة من الطلاب المعنيين. وهذه العملية تحتوى على الكثير من الخطوات اليدوية وتتطلب وقتا وجهدا و ذات تكلفة عالية، بالإضافة إلى انها غير دقيقة فى بعض الأحيان. نحاول من خلال هذا البحث أن نحل تلك المشكلة بتقديم نموذج جديد وعملى لتقويم السؤال بطريقة مباشرة وفورية عن طريق الشبكة العنكبوتية. فمن خلال هذا النموذج يمكننا أن نحصل على إستجابات الطلاب– بشكل فورى– على إختبار إستطلاعى يتم بنائه وعرضه من خلال الشبكة العنكبوتية. وبتطبيق بعض الأساليب الإحصائية على تلك الإستجابات يمكننا إستخلاص خصائص تلك الأسئلة، والتى من خلالها يمكن للنموذج إكتشاف الأسئلة التى بها مشاكل و الأسئلة الضعيفة، لتحذف من بنوك الأسئلة، ويتم تخزين الأسئلة الجيدة فقط فى تلك البنوك. كل ذلك يتم بسرعة وبدقة وبأدنى تكلفة.

1. Introduction

The fifth generation of the distance learning -Intelligent Flexible Learning Model- has very important impacts on: society, students, and institutions [27]. Making learning of all kinds, at all levels, any time, any place, any pace-a practical reality for every man, woman, and child. Equity of educational opportunity, saving time, and saving cost are provided by distance learning (DL) [25, 26]. Distance students or learners can study at their own pace and in the manner that suits their lifestyle. DL will overcome the educational deficiencies resulting of the high- density classrooms at pre-college level and in higher education, by centering learning around the student instead of the classroom. Also, it focuses on the strengths and needs of individual learners to make lifelong learning a reality. In addition, it overcomes professional teacher shortages; it offers education in places where there are no resources or where few exist; it extends the learning day and the learning place. The DL may be used as a virtual campus (alternative to the classroom setting), and may be incorporated into traditional higher and pre-college educational systems as an educational tool for students (to supplement their traditional learning experience) [9, 25].

In literature, many researchers have observed that 80%-90% of tutorial utterances were in the form of items [2, 12, 17]. Subsequently, items and their quality is one of the most important issues in distance learning systems. Poor and problematic items (PPitems) may destroy the idea of distance learning. Therefore, we should ensure the quality and integrity of the items before storing them finally in item banks, and becoming available for DL systems [1, 10, 15].

The traditional process for evaluating an item is performed in schools, by administering a paper-and- pencil pilot test, so-called tryout. A set of items is administered to a pilot sample of students similar in characteristics to the examinees for whom the test is intended. Data from this pilot study can then be analyzed to derive item characteristic indices. These indices are used to guide the revision of the item to produce a final test with maximum reliability [16]. This traditional process consists of many manual steps. It includes preparing tryout items, selecting schools, selecting classes, determining the pilot sample of students, determining the proctors, administering tryout in the schools, scoring tryout, and accommodating the data for analysis. This process is costly, very time-consuming endeavor, and sometimes inaccurate.

In this paper, we attempt to solve this problem by introducing a new practical model for evaluating and controlling item quality online. Through this model, the tryout is administered online, rather than using paper-and- pencil, to a pilot sample of students. As a result, our model can get, on the spot, student responses. Our model then performs a kind of analysis using some specialized techniques upon these responses, to identify item characteristics. Our model will use instantly these characteristics to detect weak, problematic, and/or good items. It will also detect items that need revision and those items with ambiguous distractors. All in quick and accurate manner. This model could be used by practitioners in e-learning systems for controlling the items before lunching them in their systems.

The system that puts our model into practice has been developed**.** This system is implemented using C# and Microsoft SQL server. It runs under Microsoft Windows 2003 server platform. Currently, this system is undergoing a comprehensive testing. Passing the testing stage on samples from the Egyptian preparatory schools curriculum, the system will be published on the web site of the National Center of Examinations & Educational Evaluation (NCEEE), Egypt. We will evaluate its usage in subsequent status paper once all information is available.

The rest of the paper is structured as follows. Section 2 introduces the item banks and discusses its importance within the distance learning system. In section 3, we present the proposed item evaluation model. In section 4, we describe, in details, our model and supporting mechanism for Item evaluation online. Section 5 gives some concluding remarks and directions for future research.

2. Item Bank as an Integral Part of DL

The item bank (IB) is a large collection of test items with two types of item metadata. These metadata are: *descriptive* metadata, or DSmetadata for short, and *psychometric* metadata, or PSmetadata for short. For more details see [1, 15, 16]. The main purpose of the item banks is to make the task of student assessment easier and accurate.

A good item bank has a number of distinguishing features. First, the number and type of items faithfully reflect the nature and emphases of the knowledge domain to be measured. More so than paper-and-pencil testing, online testing requires a sufficient number of interchangeable items on each test objective for multiple-form and adapted tests.

Second, the items meet accepted standards of content validity and psychometric quality. So, these items should be tested to ensure their quality before storing them finally in the IB, and before lunching them in DL systems. Simply put, the items measure what they are supposed to measure and they do so very well.

Third, the item bank is easy to use and maintain. Content specialists can easily manage the test items and build tests to their specifications [16].

The position of the item banks within teaching-learning model architecture of the DL is illustrated in Figure 1. The teaching-learning model of the DL is considered the heart of the DL. In literature, many efforts have been made to design this architecture, e.g. [3, 4, 7, 11, 18, 20, 21, 22, 23, 24]. However, these efforts did not address this architecture in the needed and sufficient details. On the other hand, they overlooked some essential components from their design, such as: item banks, student profile, learning profile, and presentation module. We have tried to propose a new design of this architecture, that includes these necessary components, and to clarify the proper location of the item banks within this architecture. This is illustrated in Figure 1.

This figure shows that the teaching-learning model consists of the four main components: Student model, Tutoring model, Course model, and Interface model.

A student model component permits the system to store relevant knowledge about the student and to use this accumulated knowledge as the basis for system adaptation to student needs [5].

Tutoring model component is the heart of the teaching-learning model; it selects problems to be given to students and generates appropriate instructional actions and decisions according to the student model [3,8].

The interface model component provides the means of communications between the learner and the system [3, 8].

The course model is structured into course material, course metadata, ***item banks***, and frequently asked questions [5, 19].

From the item banks, a DL system can draw high quality items that are matched to a specific measurement need or purpose. These systems should use item banks to generate: daily items to students, weekly quizzes, monthly quizzes, tests for each part of the course, and final tests.

Accordingly, the item bank is an inevitable integral part of DL systems.

3. Item Evaluation Model

The Item evaluation model has been designed to evaluate an item online. This model describes a novel approach for ensuring the quality and integrity of the item. We have designed this model -mainly- by exploiting our practical experience in the manual processes of the item evaluation. Although literature [1, 2, 10, 15, 16] did not discuss directly the item evaluation issues in web-based environment. However, these research works have guided us to explore some components of our model and their functions. Figure 2 shows the architecture of the item evaluation model. This architecture is comprised of six major modules:

- Student interface,
- Instructor interface,
- Tryout generator,
- Scoring,
- Analyzer, and
- Evaluator.

These modules and their functions are compatible with the item development life cycle (i.e. preparation, delivery, and evaluation). Moreover, this architecture shows the relationships between the various modules, and the main inputs and outputs to and from each module.

The data requirements -in our model- are represented by five preliminary databases. These databases store and maintain data about: the items and their metadata, tryout specs, student responses, and student profile. These preliminary databases are:

- Item bank,
- Student profile,
- Tryout specs,
- Responses, and
- Poor and problematic items.

There are two major educational measurement theories today: classical theory and item response theory (IRT), which are used for determining item characteristics and test attributes [15, 16, 28, 29]. From these theories, a large array of statistical measures and indices has been suggested as appropriate for deriving item characteristics.

We have used the statistical techniques of the classical theory in our model [1]. From these techniques, we have proposed two cooperative algorithms for controlling the item quality.

The first algorithm calculates the item difficulty index, from student responses, and inspects this value carefully to determine whether this index is in the acceptable range. If so, we apply the second algorithm. Otherwise, a decision should be taken to revise the item or eliminate it. The second algorithm is used to calculate the item discrimination index, from student responses, and inspect this value carefully to determine whether this index is in the acceptable range. If so, the psychometric metadata of the item will be stored in the item bank. Otherwise, a decision should be taken to revise the item or eliminate it.

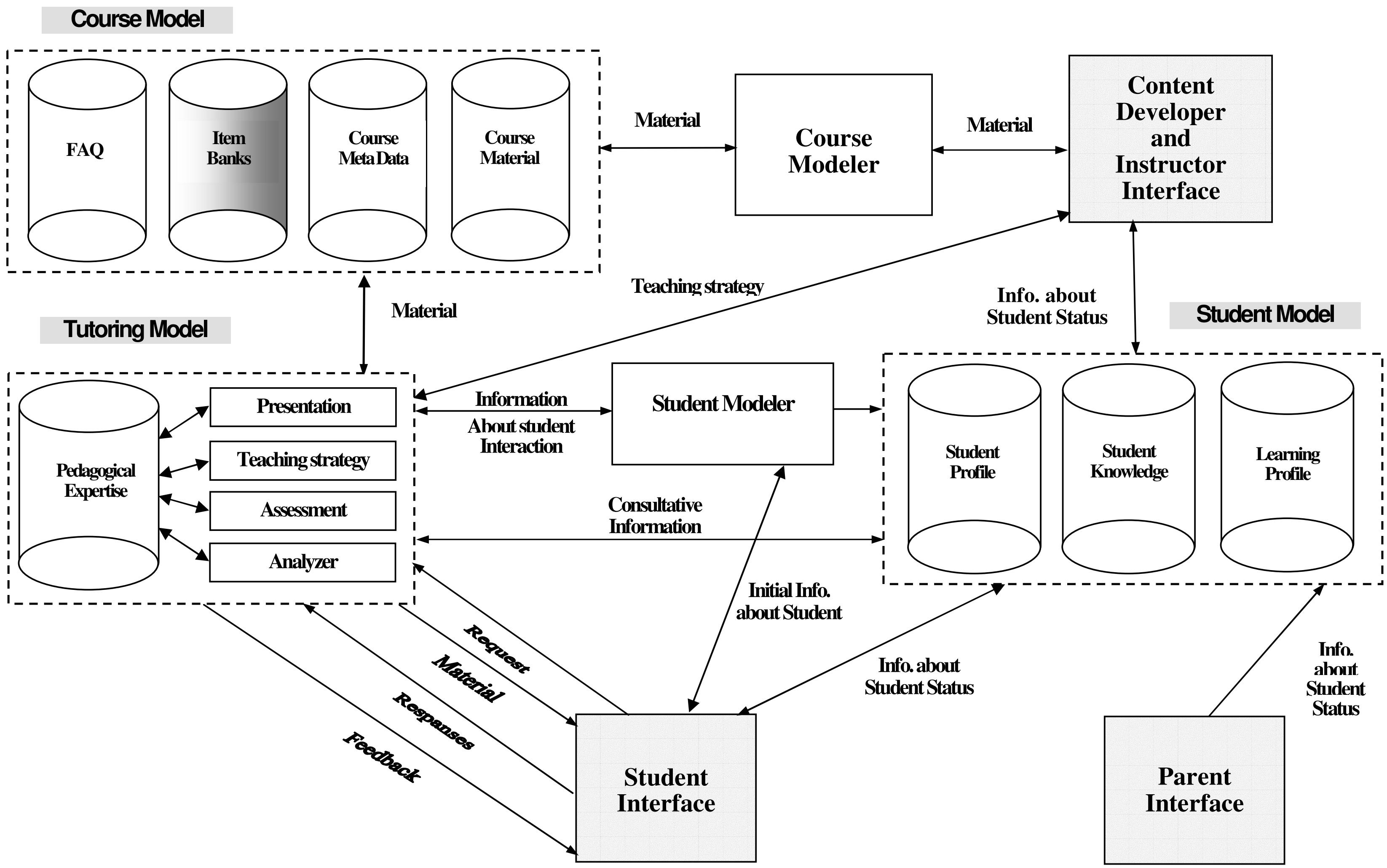

Figure 1: Position of the Item Banks within Teaching-Learning Model

Our model can be adapted easily in order to deal with the item response theory (IRT). This can be achieved by replacing analyzer module and a part of evaluator module by an IRT-based application such as: Microscale, BILOG, MULTILOG, PARSCALE2, DIMTEST, DETECT, and MicroFACT.

In the next section, we will describe the components of our model and their functions in details

4. Model Description

We have analyzed the life cycle of item evaluation in DL (see Table 1). We decomposed the life cycle of item evaluation into three stages: preparation (before tryout), delivery (during tryout), and evaluation (after tryout). Each of these stages is further decomposed into smaller stages [2]. The modules of our model and their functions are compatible with these stages. Each stage has the correspondence modules that can manage and conduct it, see Figure 2.

Table 1: Item evaluation life cycle

Preparation	Delivery	Evaluation
• Author • Review • Store item and DSmetadata	• Selection • Presentation • Getting the answer • Scoring	• Item Analysis to determine item attributes (PSmetadata) • Item Evaluation steps: 1- Discover and Eliminate PPitems. 2- Store good items in IB

4.1 Preparation stage

Life of an item begins at *authoring* time. Item and its descriptive metadata are created by human authors, instructors and content developers. Multiple choice item (our model is concerned with multiple-choice items) has the following components: the item itself (or *stem*), and a set of options (distractors and the answer). In addition, there are two types of item metadata:

1- Descriptive metadata (DSmetadata) such as: item ID, topic name, cognitive level (i.e. knowledge, comprehension, application, analysis, synthesis, evaluation), allowed time, number of attempts, difficulty level (i.e. very difficult, difficult, intermediate, easy), item answer, item mark, and feedback.

2- Psychometric metadata (PSmetadata) such as: difficulty index and discrimination index.

These items and their DSmetadata are reviewed by a committee to ensure the integrity of these items. These items and their DSmetadata are stored directly to the IB by the instructor interface module. However, the PSmetadata will be added later to the item bank after evaluation stage [1, 2, 15], see Figure 2. .

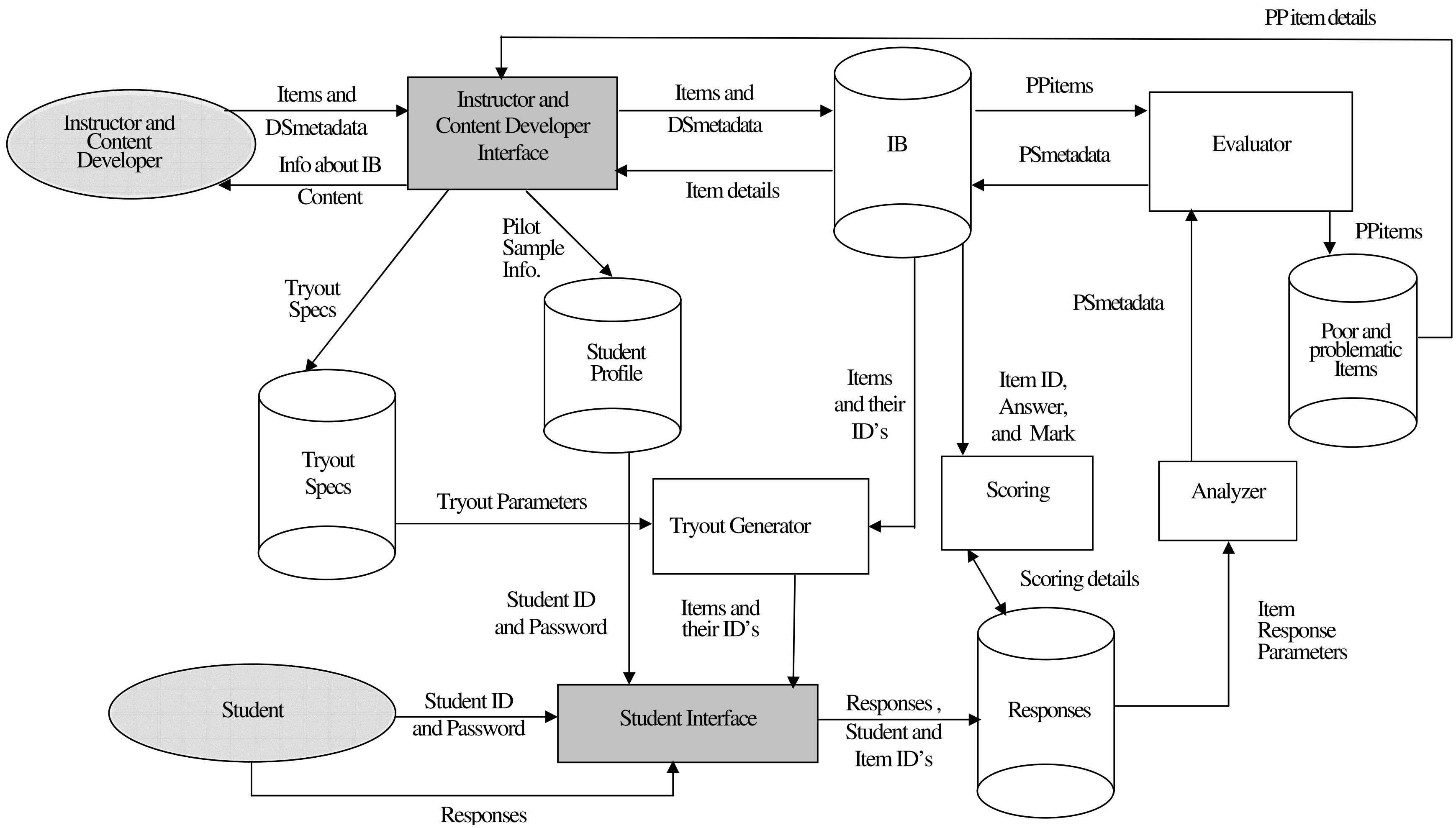

Figure 2: Item Evaluation Model in DL Environment

4.2 Delivery stage

The second stage is the item delivery to the tryout; it is the active life of the item [2]. This stage is divided into two phases: tryout building and presentation, and get answer and compute scoring.

A) Tryout building and presentation:

The active life of a stored item starts when it is selected for presentation as a part of a tryout. Content developer (or instructor) determines, in advance, a number of parameters, and store them in tryout specs. These parameters include [15, 16]:

- Stage and class,
- Subject matter,
- Unit,
- Topic where the item belong to,
- Cognitive level for each item,
- Difficulty level for each item, and
- Number of items in the tryout.

By using these parameters, the tryout generator will select the appropriate tryout items.

Student logs into the system by providing the user name and password. The student interface checks this information against the student profile. If this information is incorrect or the student tries to login more than once in the same session, then the student is not authorized to start a session. Otherwise, the tryout generator sends the following information to the client side: tryout items, item options, and item allowed time. Moreover, tryout generator will trigger student interface in order to display the items, one after another. The item stem and its options are presented to the student to answer.

After the allowed time of the item expires, the current item diminishes -in a timely fashion- and the next one will be presented. This process continues until the total number of tryout items is reached. Figures 3 through 5 illustrate snapshots of the instructor interface (tryout manager), the item presentation, and the tryout feedback.

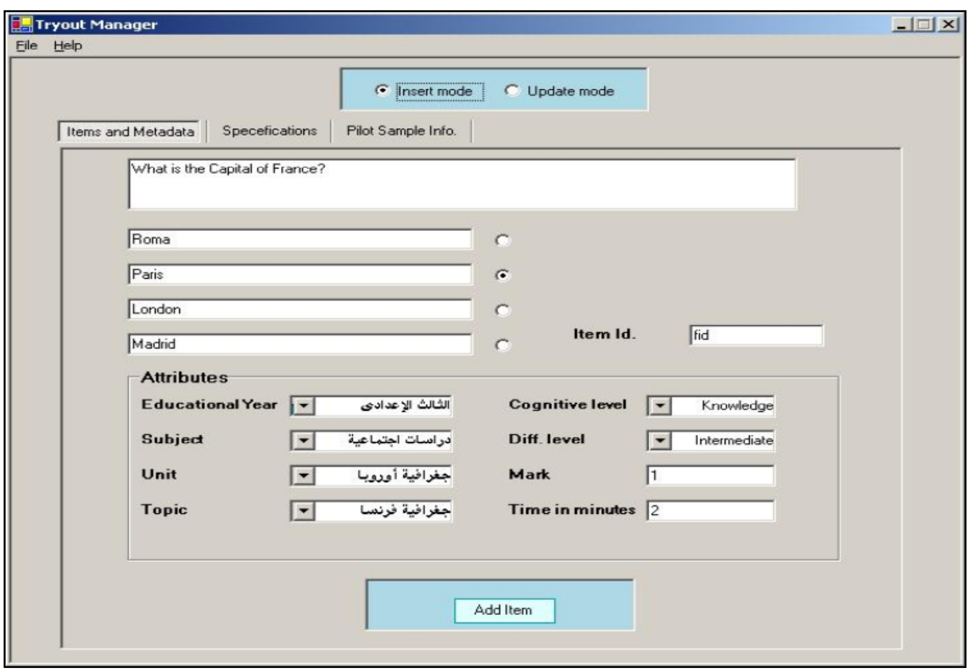

Figure 3: A snapshot of the tryout manager (item and metadata)

Figure 4: A snapshot of the item presentation

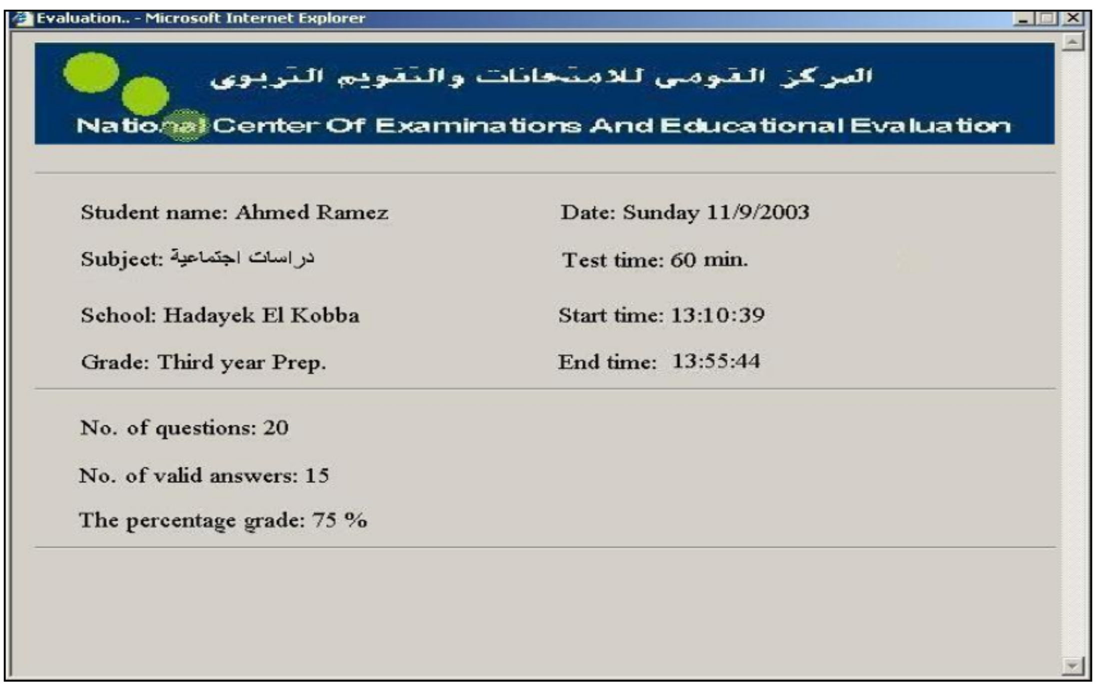

Figure 5: The tryout feedback

From our viewpoint, the security of the tryout taking is a vital challenge facing the model. What is the most secure way to handle online tryouts? There are two possible ways to apply the tryout. The first one is to perform it in secure centers or in school labs as a proctored test. The second one is to perform it at home. The latter is better in the terms of cost, but less secure. In our model, we propose the following strategy that tries to minimize a student's temptation to cheat:

1- Using student ID and password to allow only authorized students to register for the tryout, see Figure 6.
2- The student is not allowed, under any circumstances, to register more than once during the course of the same session.
3- Storing some personal and educational information about student in his profile (e.g., student ID, password, Email, phone No., mobile, general knowledge level, etc.), to monitor and control student interactions, and to compare student general knowledge level with the tryout result, see Figure 7.
4- Limit the times; ensure that the tryout is taken within a certain amount of time, and each item has a specific period of time to answer, see Figure 4.
5- There are no retries at all (i.e. back navigation is prohibited)**,** see Figure 4.
6- Preventing student from saving the items in his machine or printing them**,** see Figure 4.
7- Just after the tryout session, a sample of students is randomly selected by the system. Those students are checked by answering unannounced oral questions - by phone calls - about the topics in the tryout.

Figure 6: Student login authorization

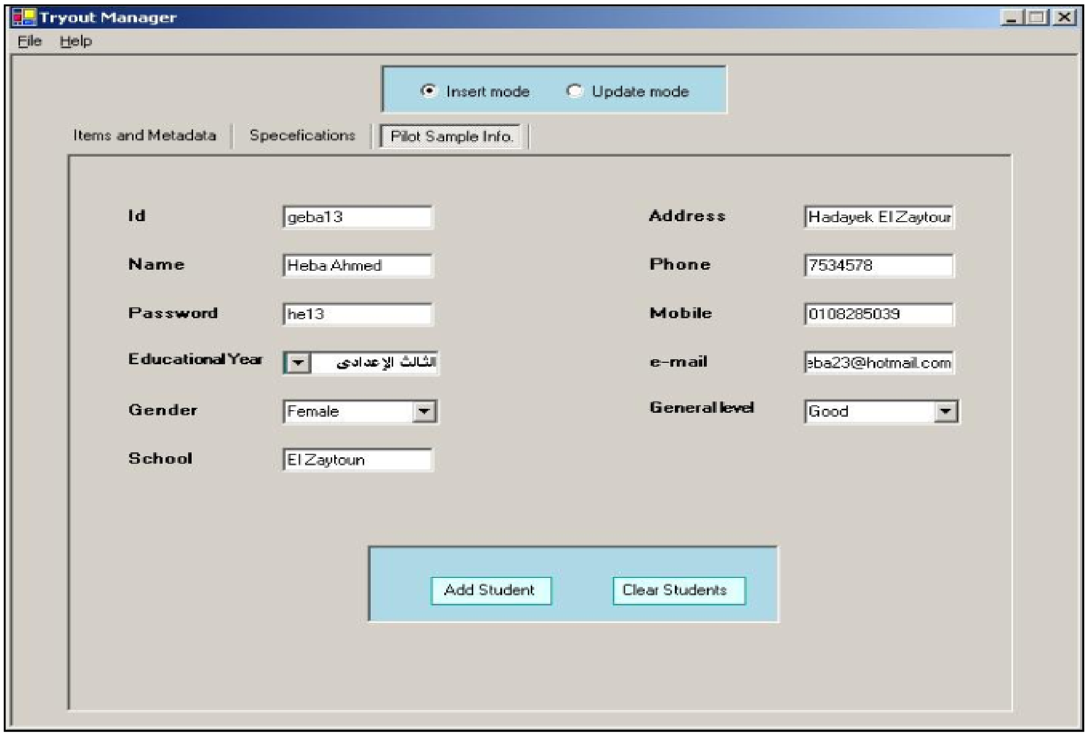

Figure 7: The pilot sample information

B) Get answer and compute scoring

During the tryout session, student interface gets the student response -for each item- on the spot. These responses are stored in student responses database. Student responses contain the attributes: item ID, student ID, student response, and student score, (see Figure 2).

Once the tryout session has been completed, the scoring module starts immediately to score the items. It gets the item answer and its mark from the IB. And it matches student response with the item answer. If both are identical the item mark is stored in student score. Otherwise, the student score will be set to zero, (see Figure 2) [14].

4.3 Evaluation stage

Evaluation stage is the third stage in the item development life cycle.

It is considered the core of our model. In this stage, the analyzer module analyzes student responses to determine item PSmetadata. The evaluator module, in turn, will evaluate these PSmetadata to discover the item quality. This stage is divided into two phases: item analysis and item evaluation, see Figure 2.

A) Item analysis

Once the tryout scoring process is completed, the analyzer module starts to analyze the student responses. It gets item response parameters from the responses database. These parameters are:

- Item ID,
- Item mark,
- Number of students in the sample,
- Number of students responding,
- Number of correct responses to the item,
- Number of incorrect responses to the item,
- Total item score across students,
- The total tryout score for each student,
- The total score for each student who passes the item, and
- The total score for each student who fails the item.

The analyzer module will use these parameters to calculate the following indexes.

1) Difficulty index [1, 13, 15, 28, 29]

Item difficulty is an index that shows the percentage of students who answered an item correctly. It is calculated by one of the following formulas according to the type of the item:

$$P = \frac{\text{Number of correct responses to an item}}{\text{Number of persons responding}} \quad (1)$$

(applied only for dichotomous items)

$$P = \frac{\text{mean item score across examinees}}{\text{maximum score of the item}} \quad (2)$$

(applied only for essay items)

The difficulty index can range from 0 to 1. It is an inverse scale, since high P values correspond to easy items and low P values correspond to difficult items.

2) Discrimination index [1, 13, 15, 28, 29]

Item discrimination means that the item is effective in separating those with high scores on the total test from those with low total test scores. It seems reasonable that if an item is a good discriminator, students with high-test scores will tend to get it correct and those with low test scores will respond incorrectly. The discrimination index commonly used is the correlation coefficient between item scores and the total test scores (or with the total scores of the rest items). This is called the item-total correlation. Of course, this correlation coefficient or discrimination index can range from -1.0 to + 1.0 (but which normally cannot reach +1 or -1). The correlation coefficient typically used is the pearson product-moment coefficient for essay items and the point-biserial correlation coefficient for dichotomous items, which is a special case of the pearson product-moment coefficient. In our model, we shall calculate the discrimination index by one of the following formulas according to the type of the item:

$$R = \frac{\overline{X}p - \overline{X}q}{S_x}\sqrt{pq} \quad \text{(Applied only for dichotomous items)} \quad (3)$$

Where $\overline{X}p$ = the mean total score for students who pass the item

$\overline{X}q$ = the mean total score for students who fail the item

Sx = the standard deviation of total tryout scores which is given by the formula:

$$S_x = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N-1}}$$

(Where x_i is the total tryout score of ith student, $\bar{x}$ is the mean of total tryout scores, and N is the number of Students in the sample).

P = the proportion correct.

q =the proportion incorrect.

$$R= \frac{n\sum xy - \sum x \sum y}{\sqrt{[(n\sum x^2 - (\sum x)^2][(n\sum y^2 - (\sum y)^2]}} \quad (4)$$

(Applied only for essay item).

(where n is the sample size, x is the score of the item , and y is the total score minus item score)

Positive correlation shows that the item is measuring something in common with the total tryout, getting the item correct predicts a higher total tryout score. This is what we would hope to find for each of the items. A zero correlation shows that performance on that item is not related to performance on the total tryout. Such an item is not a useful contributor to the total tryout and, at the very least, needs to be revised or possibly eliminated. If the item-total correlation is negative, there is a problem. This means that getting that item correct is predictive of a low total tryout score. This could only occur if the item were misleading to the better students (better in terms of total tryout scores). Items that have negative item-total correlations should be eliminated from the test [15].

B) Item evaluation

After the analyzer module calculates the PSmetadata of an item, it sends these data to the evaluator module, see figure 2. This evaluator module inspects the PSmetadata carefully to discover poor and problematic items. In the following, we describe the role of the evaluator module, using the PSmetadata, in revealing what is wrong and why.

An inspection of the item's PSmetadata can be revealing to content developer. Content developers may choose to delete or revise items, based on decisions made by the evaluator module. Often there are concepts that everyone assumes are well understood, but corresponding tryout items have surprising PSmetadata. Evaluator module should determine whether the source of the problem concerns the tryout item or the instruction, and then takes the necessary action. One reason for that is the wrong scoring key and one of the distractors was really the correct response. Another reason is that the item was ambiguous, or the learning that the instructor assumed took place did not occur. Hence, the evaluator module gives the content developer information that can guide him to improve the item.

Both the difficulty and discrimination indexes (PSmetadata) provide the data about the item that the evaluator module can use to take the appropriate decisions. In literature, suggestions of the item evaluation in terms of both difficulty and discrimination indexes have been emerged [1, 13, 15, 28, 29]. From these suggestions, we can extract and specify the following production rules for item evaluation:

IF (the difficulty index ≤ 0.25)
THEN (there may be ambiguity) **OR** (confusion in the wording) **OR** (the item (has not been covered by instruction).

IF (the difficulty index ≤ 0.25)
THEN (the item should be eliminated from the IB) **AND** (it should be stored in PPitem database).

IF (the difficulty index ≥ 0.90)
THEN (the item is very easy).

IF **(**the item is very easy)
THEN **(**it should be eliminated from the IB) **AND** (it should be stored in PPitem database).

IF (0.25 < difficulty index < 0.90)
THEN **(**the evaluator module will check the discrimination index).

IF (discrimination index ≥ 0.40)
THEN (it is acceptable item) **AND** (store its PSmetadat in the IB).

IF (0.30 ≤ discrimination index ≤ 0.39)
THEN (reasonably good but possibly subject to improvement).

IF (0.20 ≤ discrimination index ≤ 0.29)
THEN **(**marginal item, need some revision).

IF (0.05 ≤ discrimination index ≤ 0.19)
THEN (check difficulty index.).

IF (discrimination index = 0)
THEN **(**the performance on that item is not related to performance on the total tryout. Such an item is not a useful contributor to the total tryout (redundant)) **AND** (needs to be revised or possibly eliminated from the IB, and stored in PPitem database).

IF (discrimination index ≤ 0)
THEN **(**there is subtle confusion or ambiguity in the item that is misleading the better-performing students but does not affect the poorer-performing

students. A clue to the problem may be found in the incorrect responses of the better-performing students) **AND** (this item should be eliminated from the IB, and stored in PPitem database).

We have derived from these findings (i.e. production rules) two cooperative algorithms, as illustrated in Figures 8 and 9 [15].

The first algorithm calculates the item difficulty index, from student responses, and inspects this value carefully to determine whether this index is in the acceptable range. If so, we apply the second algorithm. Otherwise, a decision should be taken to revise the item or eliminate it. The second algorithm is used to calculate the item discrimination index, from student responses, and inspect this value carefully to determine whether this index is in the acceptable range. If so, the psychometric metadata of the item will be stored in the item bank. Otherwise, a decision should be taken to revise the item or eliminate it.

The evaluator module uses these algorithms to perform the following:

- Detect the accepted items,
- Store the PSmetadata of accepted items, in IB,
- Detect the poor and problematic items (PPitems),
- Eliminate PPitems from IB, and
- Store PPitems in the PPitem database.

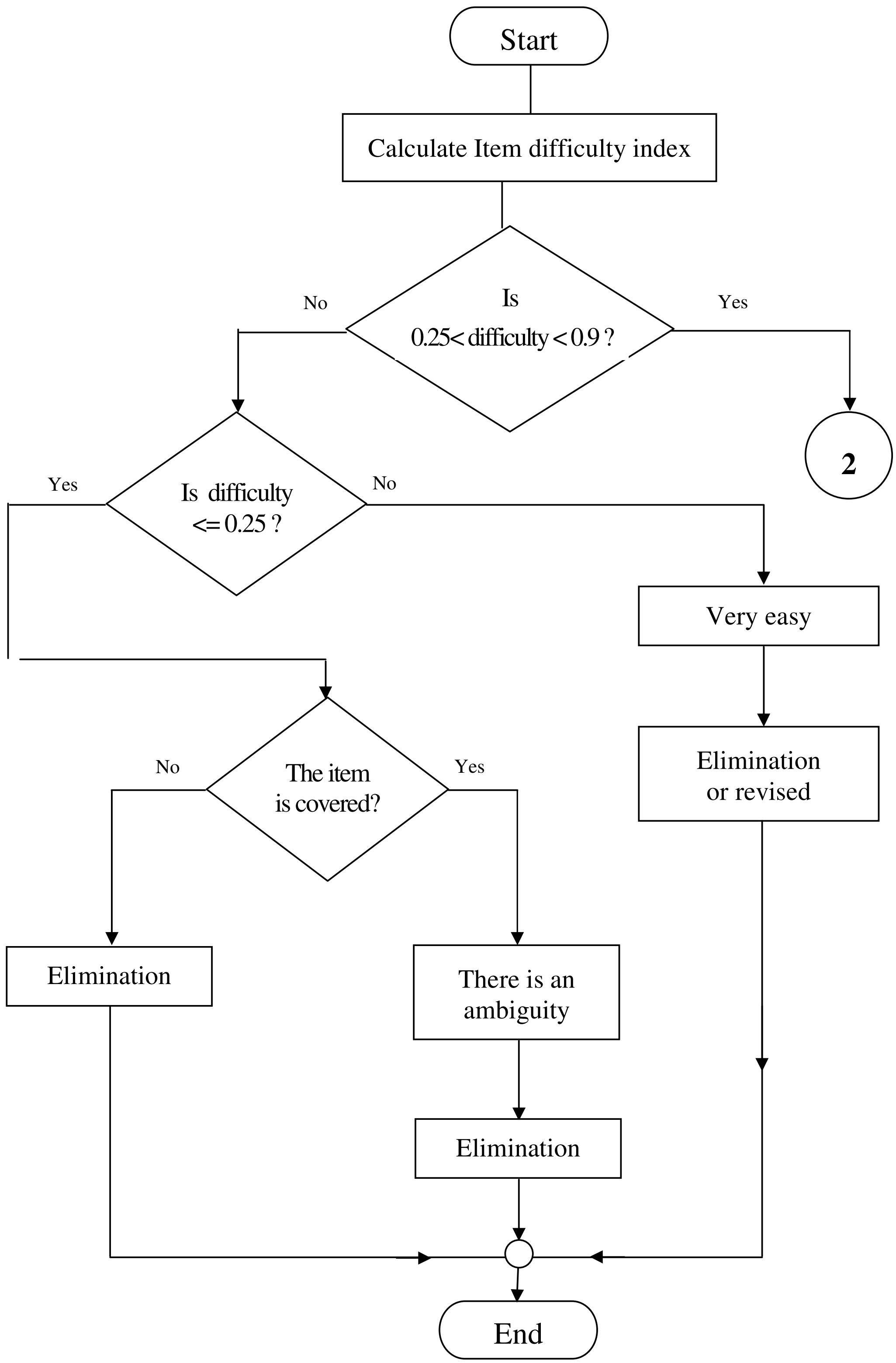

Figure 8: One item evaluation in terms of difficulty index

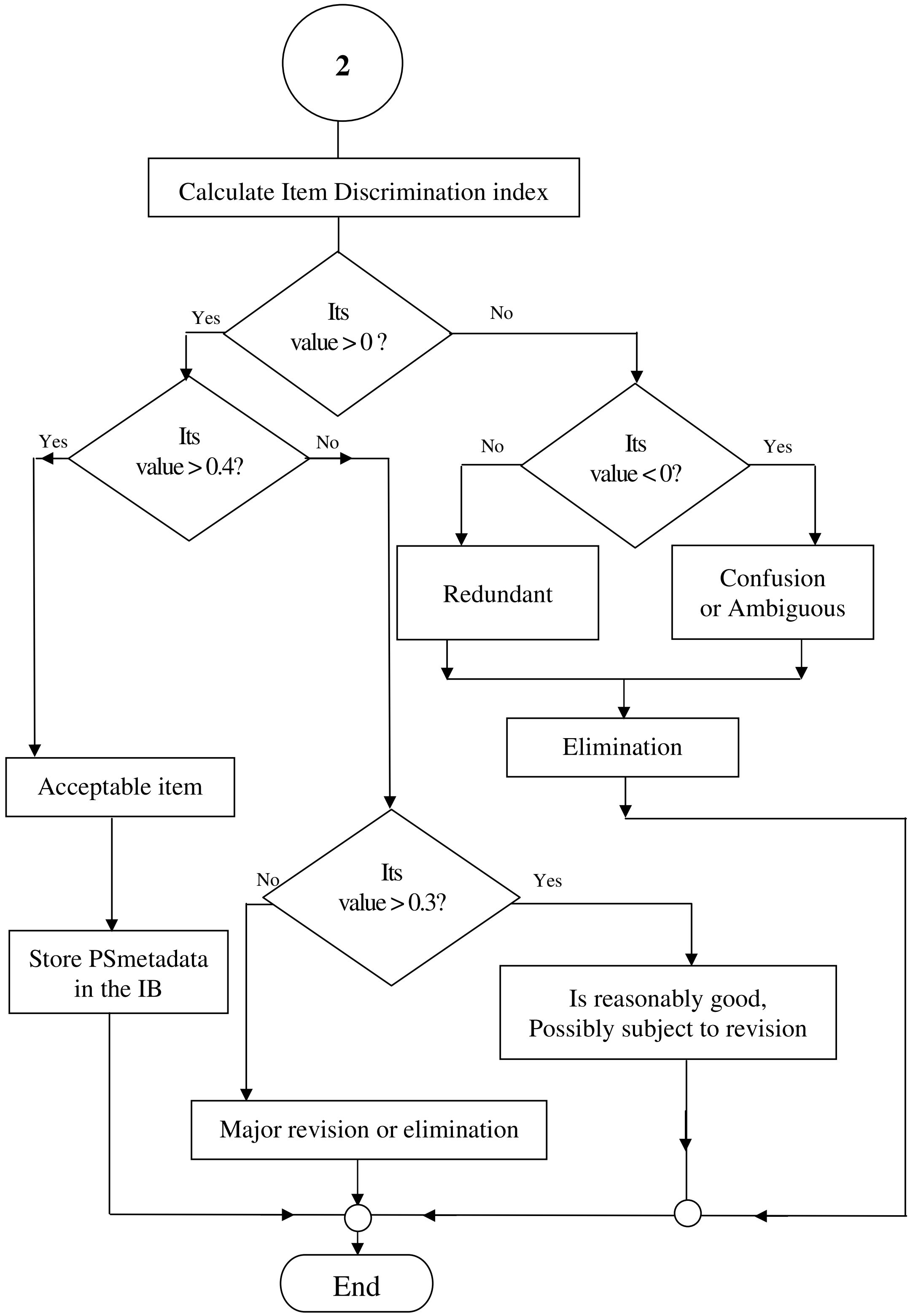

Figure 9: One item evaluation in terms of discrimination index

5. Conclusions

Items and their quality is one of the most important issues in distance learning systems. Poor and problematic items are the major obstacles facing these systems. The traditional paper-and- pencil process for controlling item quality is costly, very time-consuming endeavor, and sometimes inaccurate.

In this paper, we proposed a solution to this problem by introducing a new practical model for evaluating and controlling the item quality online. Through this model we can get, on the spot, student responses and identify item characteristics. Our new model proposed two cooperative algorithms for controlling the item quality. The first algorithm calculates the item difficulty index, from student responses, and inspects this value carefully to determine whether this index is in the acceptable range. If so, the second algorithm is applied to calculate the item discrimination index, from student responses, and inspect this value carefully to determine whether this index is in the acceptable range. If so, the psychometric metadata of the item is stored in the item bank. If the conditions for accepting any of the indices fail, the item should be revised or eliminated.

This new model could be used by practitioners of e-learning systems for controlling the items in these systems.

We also identified the importance and proper location of the item bank within teaching-learning model architecture of the distance learning framework. We have concluded that the item bank is an integral part of the distance learning systems. Item banks have important impact on distance learner assessment, from which a distance learning system can draw high quality items that are matched to a specific measurement need or purpose.

Distance learning has a crucial challenge associated with it, how do we prevent cheating while online testing? We tried to solve this problem by presenting a strategy to minimize the cheating online.

Further researches are needed to make the online testing more secure.

A potential research path includes developing a full-fledged student model for discovering and preventing the cheating on online testing. Another future research is to conduct a comparative study between the traditional method for controlling the item quality and the proposed automated approach described in this paper.

References

[1] Boshra, M., Salah, A., and Ibraheem, A. (1994). Issues in Item Banking, Item Analysis, and Evaluation of Summative Examinations in Egypt. *In the Proceedings of the Annual Conference of the Institute of Statistical Studies and Research, Cairo University, Cairo.*

[2] Brusilovsky, P., and Miller, P. (1999). Web-Based Testing for Distance Education. *In the Proceedings of WebNet'99*, *World Conference of the WWW and Internet*, pp. 149-154, Honolulu, USA.

[3] Brusilovsky, P., Schwarz, E., and Weber, G. (1996). An Intelligent Tutoring System on World Wide Web. *In the Proceedings of the 3rd International Conference on Intelligent Tutoring Systems*, ITS-96. pp. 261-269, Berlin: Springer Verlag.

[4] Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). Web-based Education for All: A Tool for Developing Adaptive Courseware. *Computer Networks and ISDN Systems,* Vol. 30, No.1, pp. 291-300.

[5] Carro, R., Pulido, E., and Rodríguez, P. (1999). Designing Adaptive Web-based Courses with TANGOW. *In the Proceedings of the Seventh International Conference on Computers in Education.* Vol. 2., Chiba, Japan.

[6] Clulow, V., and Russell, I. (2001). Online Student Assessment - A Preliminary Evaluation of Learning through Article reviews. *In the Proceedings of Australian and New Zealand Marketing Academy Conference, Massey University.*

[7] Eklund, J. (2001). Principles of e-learning development and implementation. Invited presentation to ebusiness summit, *In the Proceedings of The 2nd Annual Australia and New Zealand E-Business summit*.

[8] Eklund, J., and Brusilovsky, P. (1999). InterBook: An Adaptive Tutoring System. *Uniserve Science News*, Vol. 12, pp. 8-13.

[9] Freguson, J., Wilson, J. (2001). Process Redesign and online Learning. *International Journal of Educational Technology*, Vol. 2, No. 2.

[10] Gilbert, J. (2001). Workshop on Assessment Methods in Web-Based Learning Environments & Adaptive Hypermedia. *International Journal of Artificial Intelligence*, Vol. 12, pp. 1020-1029.

[11] Han, C. Y., and Gilbert, J. (2000). A Smart e-School Framework. *In the Proceedings of SSGR 2000: International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet,* pp. 140-148, L'Aquila, Italy.

[12] Heffernan, T. (2001). *Intelligent Tutoring Systems have Forgotten the Tutor: Adding a Cognitive Model of Human Tutors*. Unpublished Doctoral dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh.

[13] Henning, G. (1987). *A Guide to Language Testing: Development, Evaluation, and Research,* A division of Haper Collins Publisher: Newbury house Publishers.

[14] Horlad, F., Oneil, J., and Ray, S. (2003). *Technology Applications in Education*. New Jersey: Lawrence Erlbaum Associates, Inc.

[15] Ibraheem, A. (1995). Computer-Based System for Item Banks: Analysis, Design, and Implementation. *M.Sc. thesis, Institute of the Statistical Studies and Research, Cairo University, Cairo.*

[16] Lawrence, R. (1998). Item Banking. *ERIC Clearinghouse on Assessment and Evaluation*, Washington DC.

[17] Lepper, M. R., Drake, M. F., and O'Donnell-Johnson (1997). *Scaffolding Techniques of Expert Tutors, in Scaffolding Student Learning: Instructional Approaches and Issues.* M. Brookline Books, Cambridge, MA.

[18] Martin, B. (2000). Learning Constraints by Asking Questions, *In the Proceedings of the workshop on Applying Machine Learning to Intelligent Tutoring Systems*, *Design/Construction*, ITS'2000, Montreal, pp. 25-30.

[19] Mason, R. (1998). Models of Online Courses. *The Asynchronous Learning Networks Magazine*, Vol. 2, No. 2.

[20] Mitrovic, A. (1998). Experiences in Implementing Constraint-Based Modeling in SQL-Tutor. *In the Proceedings of the Intelligent Tutoring Systems*, *ITS'98*, pp. 414-423.

[21] Mitrovic, A., and Martin, B. (2000). Evaluating Effectiveness of Feedback in SQL-Tutor. *In the Proceedings of International Workshop on Advanced Learning Technologies, IWALT 2000*, pp. 143-144, Palmerston North.

[22] Mitrovic, A., and Hausler, K. (2000). Porting SQL-Tutor to the Web. *In the Proceedings of The Intelligent Tutoring Systems*, *ITS'2000, workshop on Adaptive and Intelligent Web-based Education Systems*, pp. 37-44.

[23] Mitrovic, A., and Ohlsson, S. (1999). Evaluation of a Constraint-Based Tutor for a Database Language. *Int. J. Artificial Intelligence in Education*, Vol. 10, No. 3, pp. 238-256.

[24] Piguet, A., and Peraya, D. (2000). Creating Web-integrated learning environment: An analysis of WebCT Authoring Tools in Respect to Usability. *Australia Journal of Educational Technology*, Vol. 16, No. 3, pp. 302-314.

[25] Sherry, L. (1996). Issues in distance learning. *International journal of Educational telecommunications*, Vol. 1, No. 4, pp. 337-365.

[26] Stenerson, j. (1998). System Analysis and Design for a Successful Distance Education Program Implementation. *Journal of distance learning administration*, Vol. 1, No. 2.

[27] Taylor, J. (2001). Fifth Generation Distance Education, *In the Proceedings of the 20th world Conference on Open learning and Distance Education*, *International Council for Open and Distance Learning, ICDE,* Usseldorf, Germany.

[28] Thorandike, R., Cunningham, G., and Hegen, E. (1991). *Measurement and Evaluation in Psychology and Education.* New York: Macmillan Publishing Company.

[29] Wiersma, W., and Jurs, S. (1991). *Educational Measurement & testing.* Boston: Allyn and Bacon.