Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

Journal of Software Engineering 4 (4): 288-296, 2010
ISSN 1819-4311

Re-engineering of Multiple-choice Exam-form Production Tools: Cost-effective and Quality-assurance Approach

[1]S. Bani-Ahmad and [2]A. Audeh
[1]Faculty of Science and Information Technology,
Albalqa Applied University-Main Campus, Salt, Jordan
[2]Faculty of Education, Yarmouk University, Irbid, Jordan

Abstract: This study aims at designing and implementing an exam-form generation software tool (named ExPro) for Multiple-Choice-Based (MCB) exams. The study is motivated by the fact that student number in Jordanian universities is continuously growing at high rate. This growth is not accompanied by an equivalent growth of educational resources (instructors, labs, etc.). A result of this situation is having large number of students in class-rooms. Consequently, providing and using online-examining systems could be intractable and expensive. Alternatively, paper-based MCB tests can be used. The design and evaluation of ExPro is done by considering a basic set of design principles that are based on a list of identified Functional Requirements (FRs). Deriving those FRs is made possible by developing ExPro using the Iterative and incremental model from software engineering domain. We show that ExPro proves helpful to instructors in preparing multiple-choice tests. Further, ExPro makes archiving previous exams possible and effective to search for in future. The ExPro is available for free over the Internet and has been in use. ExPro users agree upon that ExPro (1) is easy to be learned and used and (2) proves to be a cost-effective alternative to online examining systems.

Key words: E-learning, multiple-choice questions, tests, evaluation, measurement, computer-based testing, paper-based testing

INTRODUCTION

Student numbers in Jordanian Universities are continuously growing at alarming rates (MOHE). Statistics from the website of the Ministry of higher education in Jordan (MOHE) show that this growth in student numbers is not accompanied by an equivalent growth of educational resources such as instructors, labs and class-rooms. The direct result of this situation is having large numbers of students in class-rooms. This in turn limits the space available for students when setting for evaluation exams where space is vitally required to reduce cheating. Reducing cheating is critically required as studies show that more than fifty percent of students copy answers from others during exam sessions (Run-Xian and Lao-Pin, 2007). Cheating results in having unrealistic exam grades that falsely reflect student's achievement. Consequently, the quality of the education system is significantly declined and overtime, cheating corrupts the minds of students (Run-Xian and Lao-Pin, 2007).

Another source of danger threatens the education system comes from the fact that the number of instructors grows at low rate that cannot balance the high growth-rate of student

Corresponding Author: S. Bani-Ahmad, Faculty of Science and Information Technology, lbalqa Applied University-Main Campus, Salt, Jordan

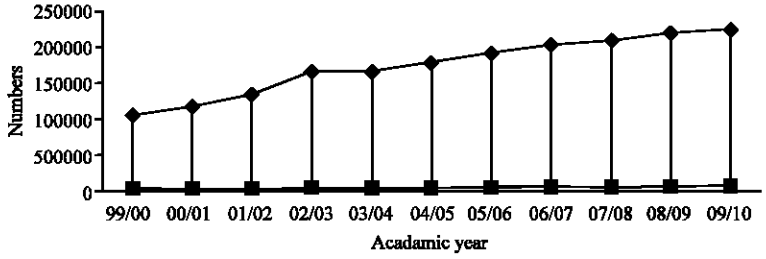
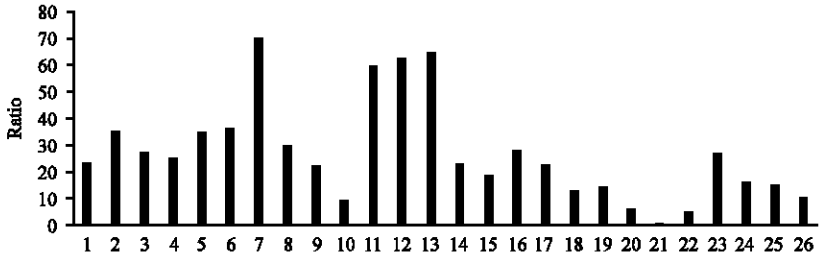


Fig. 1: Change in student and instructor numbers in Jordanian Universities from 1999/2000 to 2009/2010 (Bachelor students only)



1 The University of Jordan	10 German Jordanian University	19 Jerash Private University
2 Yarmouk University	11 Al-Ahliyya Amman University	20 Princess Sumaya Uni. For Tech.
3 Mu'tah University	12 Applied Science Uni. (Private)	21 Jordan Academy of Music
4 Jordan Uni. of Science and Technology	13 Philadelphia University	22 Educational Sciences Faculty
5 The Hashemite University	14 Al-Isra Private University	23 Jadara University
6 AL al-Bayt University	15 University of Petra	24 Jordan Applied University
7 AL-Balqa' Applied University	16 Al-Zaytoonah Private Uni. Of Jordan	25 Middle East Uni. For Graduate Studies
8 AL-Hussein Bin Talal University	17 Zarqa Private University	26 Amman Arab Uni. For Graduate Studies
9 Tafila Technical University	18 Irbid National University	

Fig. 2: Student-to-instructor ratios in Jordanian Private and Public Universities (Bachelor students only)

numbers. For instance, Fig. 1 shows the change in student and instructor numbers in Jordanian Universities over the academic years from 1999/2000 to 2009/2010 considering bachelor students only. It is clear from the curve that the number of students in Jordanian universities keeps continuously growing at high rate compared to growth in the number of instructors.

As a result of that, the student-to-instructor ratio becomes larger over time. As an example, the number of faculty members in Jordanian universities was 7600 in 2009 (MOHE). Given that the number of students enrolled in Jordanian universities for the same year was 220,000, we find that the student-to-instructor ratio is 29-to-1. This is in average. However, in some universities as shown in Fig. 2, this ratio is around 70.

High student-to-instructor ratio forms a heavy load for instructors to keep track of how students progress. Further, this makes the evaluation process much more difficult for grading process becomes exhaustive and time consuming. Instructor subjectivity while grading considerably degrades as he/she scan more and more answer sheets due to exhaustion.

Instructors find it more convenient for themselves and more fair to students to use MCB tests instead of FRB tests. MCB tests received great attention in literature. Studies show that if instructors are well trained to prepare MCB tests and the quality of items is assured, MCB exams can be an effective assessment technique (Case and Swanson, 2003; Beckert *et al*., 2003).

MCB exams are of two types, namely; Computer-Based (CB) and Paper-Based (PB). In CB MCB tests each student views and directly responds to test items (or questions) using a computer. Students' responses along with the list of questions are maintained in a computer server. This makes grading exams to be fully automated. In PB MCB tests the student uses a pencil to respond to a printed form of the MCB test. This means that computers are not required during exam sessions. Consequently, PB MCB tests are the best choice when having large number of students setting for exams. In fact, in such circumstances, CB MCB tests can be prohibitively expensive.

Another reason why PB tests are preferred over CB tests is that the test taker's prior computer experience may affect their performance; computer unfamiliarity is found to be related to lower test performance (Akdemir and Oguz, 2007).

MCB tests are more difficult to prepare than FRB tests for many reasons:

- Instructors need to prepare a large number of questions for each test as the time required to respond to free-response questions is longer than that required to respond to multiple-choice items (Burton *et al*., 1991). Thus, for an exam with a given duration of time, large number of multiple-choice questions are required to cover that duration
- Multiple-choice questions are more difficult to construct as they are relatively very sensitive to construction mistakes and, thus, require much of the instructor's focus (Burton *et al*., 1991)
- To prevent performance interference, that is cheating, MCB tests need to be produced in multiple printable forms (by re-ordering items, choices, or both)
- Each MCB exam-form requires an answer-key to be produced. Given the above three points, preparing MCB tests becomes more and more difficult (if this is manually conducted)
- The exam-forms are to be produced in printable format. The reason for this requirement is that, in the case of having large number of examiners and limited resources, paper-based exams are preferred over computer-based exams

Those difficulties may drive instructors away from MCB tests although MCB testing is the best evaluation tool in case of having large numbers of students and limited resources as we discussed before (Burton *et al*., 1991).

The goal of this study is to design and implement a multiple-choice-based exam-forms production tool that is capable of automatically producing multiple forms out of a given set of multiple-choice questions. We refer to this tool as the ExPro (Exam-forms's Production tool).

PROBLEM STATEMENT AND FUNCTIONAL REQUIREMENTS OF THE EXPRO

The problem statement of ExPro is summarized as follows. Given a set of m MC questions Q (m). We would like to design a software tool that produces n different MCB exam-forms out of Q (m). This should be done by permuting (1) the list of items, (2) the list of choices of each item, or (3) both. Each of the n exam forms is to be produced with its

answer-key. The list of choices is to be properly permuted such that the probability of observing the correct answer at one position within the list of choices is equal to the probability of observing it at any other position in the list. The goal of this requirement is to minimize the chance that an examiner receives high score solely by clustering his/her answers at one given position, not because of his/he achievement.

The main design principles of ExPro are derived from the Functional Requirements (FR) that instructors usually face when preparing MCB tests. Those requirements have been derived through getting feedback from current ExPro users who have been using ExPro for more than a year. Deriving those FRs is made possible by developing ExPro using the Iterative and incremental development approach from software engineering domain.

The basic idea behind iterative enhancement is to develop ExPro incrementally, allowing us to take advantage of what was being learned during the development of earlier versions of ExPro. Learning comes from both the development and use of the system.

The functional requirements can be summarized as follows. Notice that each of the following functional requirements is numbered for future referencing.

- Management of multiple-choice-based tests: including creation and archiving. ExPro should help instructors properly assign names to exam file in order to facilitate referring and searching for these files in the future. This is better than having piles of hardcopies of exams and having large number of disorganized computer files, probably of different styles and file formats
- Production of exam forms to minimize performance interference during exams sessions. Manually preparing exam-forms by randomly permuting questions and their choices is time and effort consuming. For that we need to take into consideration that space available to students while setting for exams is not enough to prevent cheating. And even if large space is secured, many students have developed strategies through which they can cheat. That is why one basic requirement of ExPro is to automatically produce many-enough exam-forms to reduce cheating. Having enough exam-forms is proven to be effective toward minimizing or even eliminating cheating

One note is that, the produced exam-forms are to consist of the same set of questions in order to maintain an acceptable level of fairness when evaluating examiners.

- Manually preparing answer keys and answer sheets to exam forms requires long time and effort and is prune to error. This error propagates to affect the grading step. Thus, one basic requirement of ExPro is to be able to automatically produce an answer-key and an answer-sheet to each exam-form
- The reusing of previously created tests, where piles of test papers are stored in boxes (may be damaged or lost), or located in different locations physically or computerized, which may take long time to be found. ExPro should also facilitate archiving previous exams for future referencing and reuse
- Confidentiality and personality of tests, especially if more than one user were using a shared computer or if tests are accessible through computer network. ExPro should allow the instructor to maintain his/her exam on his/her personal computer
- Tests can be of any language. ExPro has to support exams in any natural language. In fact, for ExPro to be universal and prove useful to any instructor, ExPro should be language-independent

DESIGN PRINCIPLES OF EXPRO

The above challenges form the basis for the design principles of ExPro that can be summarized as follows:

- Providing an automated process to produce tests and answer sheets in printable format. The produced exam-forms must be optimized to combat cheating as cheating brings great harm to the evaluation process and, in turn, to the educational system (Run-Xian and Lao-Pin, 2007). Reducing cheating can be achieved through having many-enough forms produced by randomizing choices, questions, or both. The randomization should produce uniformly distributed correct-choices over the set of possible positions within choice-lists as discussed earlier. One thing to notice here is that some choices should not be permuted, e.g., All above and None of above that should remain in the last position of the choice-list. One design principle of ExPro is to provide a user-friendly mechanism to mark such choices in order to properly deal with them when generating exam-forms. One more thing to notice is that ExPro has to be able to support many-enough choices to each question in order to minimize cheating and to minimize the effect of guessing (Wise and De Mars, 2010)
- Providing an easy-to-use computerized solution to produce multiple exam forms. ExPro should be built with the "keep it simple" principle in mind because it will not be exclusively used by computer specialists, but by instructors in other fields of study who might have only basic computer skills. This means that ExPro should be written to work in Microsoft Windows environment as it is the most widely used operating system nowadays by naive computer users (W3Schools)
- To facilitate saving and maintaining tests for future uses in an organized way. ExPro should help users give expressive names to the exam files. Furthermore, ExPro should have all exam parts be saved in a single file and not to scatter exam parts over multiple files of different formats. Exam parts include (1) the set of multiple-choice questions, (2) the non-text objects such as the Fig. 1 and 2 the set of free-response questions (if any) along with the typical answers to those FRB questions, (4) and the exam information such as the course name, the semester, the date and time when the exam was/is to be held. Furthermore, ExPro should provide a way to use such information and incorporate it to formulate a proper file name for the exam.
- To maintain an acceptable level of privacy and personality of test items prepared by instructors. This can be achieved by having tests to be stored at the personal computer of the instructor rather than on a shared computer. This means that ExPro has to be a desktop (or laptop) application and not a network application
- Exams written and the exam-forms produced by ExPro, as a computer software, have to be encoded using a universal encoding scheme. Fundamentally, computers deal with numbers only. Each character is assigned a number. There are hundreds of different encoding systems for assigning these numbers (Unicode). However, no single encoding could contain enough characters. Further, conventional encoding systems may conflict with one another; that is, two different characters may map to the same number (or code), or use different numbers (codes) for the same character (Unicode). To solve this problem ExPro uses the Unicode encoding scheme to store and manipulate exams. Unicode enables ExPro to be targeted across languages without re-engineering. It allows for archiving exams without corruption

- Exam-forms can be of any natural language. Thus, ExPro should support both patterns of drawing text; namely, (1) the right-to-left and (2) the left-to-right patterns. The Greek alphabet and its successors (e.g., English language) have settled on a left-to-right pattern. Other scripts, such as Arabic and Hebrew, came to be written right-to-left. ExPro should take this observation into consideration
- One important design principle of ExPro is to have continues technical support and to smaintain a point of contact with ExPro users. This is important for two reasons, (1) first, to inform users with the latest software update bug fixes and (2) second, to receive feedback and suggestions from users on how to make the software more effective and user-friendly. This can be achieved via providing an *Internet forum*, or a *webgroup page,* which is an online discussion site

MCB EXAM-FORMS PRODUCTION (MCBEP) TOOLS

Two types of MCBEP software exist, namely; online (computer-based) and offline MCBEP packages. Online MCBEPs produce online exam instances viewable on multiple personal computers. Each examiner is assigned one dedicated computer to view and answer the exam (Bani-Ahmad and Audeh, 2010). Usually, such exams are saved on computers in formats viewable by web-browsers through the Internet or intranet. In each examination-session, the items (or questions) of the exam are answered by students and graded online (on a server computer) (Bicanich *et al*., 1997).

Using Google search engine to search for software tools on multiple-choice exam bring relatively long list of related pages. Next is a list top Google-scored online MCBEP software tools returned by Google (which might be many): (1) The Multiple Choice Quiz Maker (Tac-Software). (2) (3) The ExamBuilder (ExamBuilder). Those software packages are not designed and optimized to (1) produce printable exam forms and (2) produce key-answer table for answers. The goal of all these packages is to produce interactive and computer-based tests viewable by web-browsers. Further, many of these solutions are not easy to use by naive computer users.

Offline MCBEPs produces partially computerized MCB exams as follows: the instructor enters the exam questions and the MCBEP software produces exam forms in printable formats. The printed exam-forms are answered by students. Answered exams are then manually graded by the instructor.

Offline MCBEP tools are much less in number than online MCBEP. An example of offline MCBEP software is Schoolhouse Test (Schoolhouse). It offers the ability to produce tests with various types of question and answer sheets. In addition, it provides the ability to store the inserted questions and actual test documents for future use. However, Schoolhouse test does not produce answer keys to the exam forms it produces. Further, the instructor need to manually reorder questions in order to produce multiple forms of the same exam (Schoolhouse).

EVALUATION OF EXPRO

The ExPro, as a research project, has started back in 1999 when the first version of ExPro was launched and put into service. The second version of ExPro has come to existence on April 2009. Through the testing period, ExPro has faced several enhancements. A Beta version of this solution has also been put into limited service to obtain feedback from users

before the announcement of the final working version, which is version 2.8 which is available for free download at the project's website (ExPro). The main contact point between the ExPro developers and the ExPro users is the ExPro webgroup page.

ExPro is easy to install and use with no need for programming knowledge by users. The basic usage of ExPro to prepare a multiple-choice exam involves the following main steps: (1) entering the list of multiple-choice questions (items) of the exam, (2) entering the information that will appear at the header of the exam's final forms and (3) entering the list of figures referenced in the list of exam items.

Producing exam forms starts by pre-parsing the provided list of questions to check for any structural errors and then producing the required number of exam forms along with the answer key of each exam-form produced.

The following is a summary of ExPro currently-supported features (in version 2.8 of the software):

The capability of saving exam, with all its components, for future referencing and reuse into one single, portable, ExPro file on the personal computer of the instructor. The file includes the list of questions, its answers, exam's meta-data, figures, equations etc. This feature is oriented toward design principles.

The capability to produce up to 20 different and compact (two-column) exam forms by automatically and randomly reordering the provided list of questions and their list of choices. The produced exam forms are in printable Microsoft Word (.doc) format. The user is free to write the exam in right-to-left or left-to-write languages and using any character set as characters are saved in Unicode form. Those features are oriented toward design principles.

Supporting directives or tags to enable or disable the reordering of individual choices in the list of choices of a given question. This feature is needed to support the choices of the form none of above, or both [A] and [C]. This feature is oriented toward principle.

Support of free-response questions in addition to multiple-choice questions. It also supports entering non-text objects that represent figures, equations, tables, charts, etc. This feature is oriented toward design principles.

Exam files' organization. ExPro helps in organizing exam files for future referencing by automatically assigning proper and meaningful names to the source ExPro exam-files as well as the produced exam-forms. This feature is oriented toward design principle.

Support of multi-response multiple-choice items where more than one correct answer is possible. The student responds to such question by choosing the best choice, or all correct choices (Case and Swanson, 2003). This feature is oriented toward design principle.

Facilitating exam grading (counting correct responses) of the multiple-choice part of the exam. Later in this report we will be presenting the mechanism that quickens this process through the use of transparencies. This feature is oriented toward design principles.

A typical ExPro exam file contents and the structure of MCB items can be found in (ExPro).

After the instructor chooses the required number of exam forms, ExPro pre-parses the feed test items to check the inserted questions for any structural error. If no errors were found, the forms can then be produced.

Prior to producing exam-forms, the user selects (1) the language and the writing-direction pattern (right-to-left or left-to-right), (2) the number of exam-forms required. And (3) the permuting options; that is; whither the user would like to produce exam-forms by permuting choices, questions or both. Furthermore, the user decides to show or hide the exam-form identification number. Hiding the exam-form-ID is recommended to reduce cheating.

To further post-process and print the generated exam-forms, the exam forms are saved into. doc format which is editable through Microsoft Word. Figures and attachments are saved in a separate file.

Toward the design principles, the ExPro exam file saves the questions, the exam header, the figures and the exam metadata. ExPro automatically suggests a proper file name of the exam.

To practically evaluate ExPro, we have put it into service for 10 months. During this period, feedback from users is taken and the software interface and features are modified and enhanced accordingly. Three approaches are utilized to gain feedback from users: (1) Personal contact between the users and the author. (2) Web-based feedback forms through the website of the ExPro project. (3) Through workshops in the Academic Development Center at Jordan University of Science and Technology and in the Faculty Development Center at Yarmouk University.

CONCLUSION

In this study we presented ExPro, a tool that enables the educator (e.g., teacher) to construct his/her own multiple-choice-based tests in multiple forms out of the same set of items. ExPro has unique features that make it superior to other available packages in the market. For instance, ExPro is capable of producing styled printable multiple exam-forms. For each exam-form, ExPro produces an answer sheet and an answer key. This significantly reduces the effort required by the educator to prepare multiple-choice based exams. ExPro proves to be a cost-effective alternative to computer-based examination systems while maintaining the same level of quality assurance in terms of cheating prevention.

ABOUT THE EXPRO PROJECT

The ExPro Project has initially started in 1999 and resumed in July 2008. The goal of ExPro was to design and implement a Windows-based software solution that helps instructors to prepare and manage multiple-choice-based exams. The primary design principle of ExPro is to facilitate producing exam-forms out of a given set of multiple-choice questions with minimal efforts. For more information about this project please visit the project's website (http://sites.google.com/site/theexprosite).

REFERENCES

Akdemir, O. and A. Oguz, 2007. Computer-based testing: An alternative for the assessment of Turkish undergraduate students. Elseiver Comput. Educ., 51: 1198-1204.

Bani-Ahmad, S. and A. Audeh, 2010. The ExPro 2.8 documentation. http://sites.google.com/site/theexprosite

Beckert, L., T.J. Wilkinson and R. Sainsbury, 2003. A needs-based study and examination skills course improves students performance. Med. Educ., 37: 424-428.

Bicanich, E., T. Slivinski, S.B. Hardwicke and J.T. Kapes, 1997. Internet-based testing: A vision or reality. J. Technol. Horizons Educ.

Burton, S., B. Sudweeks, R.P. Merril and B. Wood, 1991. How to prepare better multiple-choice test items: Guidelines for university faculty. Brigham Young University Testing Services. http://testing.byu.edu/info/handbooks/betteritems.pdf.

Case, S.M. and D.B. Swanson, 203. Constructing written test questions for the basic and clinical sciences. National Board of Medical Examiners, Philadelphia, PA.

Run-Xian, Z. and Lao-Pin, 2007. On the cause of university students cheating phenomenon from theperspective of Albert Banduras reciprocal determinism. US-China Educ. Rev.

Wise, S.L. and C.E. De Mars, 2010. Examinee noneffort and the validity of program assessment results. Educational Assessment, 5: 27-41.