

Ontological analysis of a Computer Based Assessment Developer fit for the School of Medicine

Bee Sung Kam¹, Il Kon Kim²

Abstract

We designed and implemented a computer based assessment developer tool fit for the school of medicine. Issues about the implementation include the ease of usage, being user friendly and convenient conversion of PBT (Paper Based Test- existing exam items) material to CBT (Computer Based Test). Assessors with a little drag and drop can create exam papers in no more than a minute.

The unique format of assessment upload, with related conversion and some mistake correction covered by automation of ontological relations are special advantages of this implementation. What makes this architecture well designed for the school of medicine is that the software techniques being used to handle upload and download of large size files (e.g. medical images). A large population can be assessed using this method without hindering the service.

Keywords: Ontology, CBT (Computer Based Test)

Introduction

Computers are regularly used to deliver, mark, and analyze student assessments (Seale, 2002). They have become an established means for this purpose (Appel, 2002, p.113). Computer Based Assessment (CBA) is a convenient and flexible alternative to traditional Paper Based Assessment and a method for delivering examinations with more statistical information. However, the development of a comparable Computer Based Test (CBT) for student assessment involves some challenges. Among the challenges are issues of test security, test assembly, test delivery, data management, and score comparability (Choi, 2002). The focus of the present research was to enhance the flexibility of tools that digitalize test assembly for administrators.

Although many CBT tools are available (online or offline) their low competency to convert existing materials in the process of item assembly hinders recommendation of such tools by test administration.

Most tools rely on users design and input, without taking any account of mistakes the user may make, or the amount of time that the assessor consumes to copy and paste questions one by one.

Internet-based item banks have been discussed by many researchers as a solution to item creation to create, manage and deliver electronic competence assessments without developing each item individually (Plichart, 2004). However, where the item banks come from is mostly ignored as it leads back to the original paper based materials (books, past exams, handouts etc.) that are not based on the item banking format.

Despite efforts in area of CBT, it is obvious a test administrator cannot present a computerized version of a traditional test in the same way. Instructions must be written specifically for the computerized version, examinees must be familiar with operating a computer well before taking a computerized exam, and the administrator must be able to deal with computer specific problems during the test (Bugbee, 1996).

Many research papers focus on solving this problem by finding ways to minimize item usage: expanding the number of test items in a bank; by hiring extra item writers and/or using item generation forms and algorithms, (Pitoniak, 2002), establishing conditional item

¹*Pusan National University Medical Education Unit, Republic Of Korea
E mail: beesung@pnu.ac.kr*

²*School of Computer Science, College of IT Engineering Kyungpook National University, Deagu, Korea
E mail: ikkim@knu.ac.kr*

exposure controls (Revuelta 1998), rotating item banks, expanded initiatives to reduce sharing of test items on the internet and better item bank utilization (Revuelta 1998; Yi,Q 2001).

In reality, what happens is the hiring of extra item writers and a typist to digitalize written items from paper based material into the computerized format of the item banking system. As a result, one must be prepared to involve cost, time and effort for assembling items to be used for assessment as Computer Based Tests.

Thus, lack of conversion tools to convert raw materials to computerized versions has made the assessor's job very difficult, tedious and time consuming. This is a worldwide limitation of usage of CBTs.

The purpose of this paper is to explore online assessment, considering convenience of conversion for assessors in case of existing raw material for CBT. Minimizing time of copy and paste and obtaining some quality of intelligence in software automation is the focus of this paper.

The first stage of developing this technology started in late 2006. This paper is the first work in this regard, and intended to solve a portion of user interface problems that have not yet been taken into consideration by common Computer Based Test products.

Background

Question design

Early computer based tests were restricted to objective, text based questions and answers (Thelwall, 2000). Plenty of software for creating variety of online assessments is widely available. Assessors can, for example, incorporate images, drawings and multimedia into questions, thus increasing the potential for testing beyond that offered by traditional paper based tests. The Tripartite Interactive Assessment Delivery System (TRIADS) resource at the University of Derby also has several of the different question types that can be designed with modern software; free software is also available to develop computer based tests at University of Leicester's CASTLE website (www.le.ac.uk/castle). Several commercial software packages (such as Question mark) are also available, that allow professors to create and operate computer based testing with minimal training.

Challenges of Question Design in Medical Schools

However the more challenging portion of question design is that an assessor would have to type, copy or paste the question line by line, add descriptions to the question, attach image files or tables or related symbols one after another and add multiple questions in sequence. At this point, the assessor has finished designing only one question. The same task should be repeated for designing further questions, which in assessor's point of view is tedious and time consuming. In medical schools, professors have many important responsibilities, with minimum time in his schedule. Thus, consuming time for the above process is unacceptable for IT technology in the 21st century.

As a computer engineer it is not amazing to leave the job for the assessor and merely readjust the design to be termed "Computer Based Test", although a computer can do better and handle more difficult tasks. The reason why computer technology has not yet penetrated into medical field is the tedious and time consuming operation of software although implementers are proud of developing it.

We designed and implemented a computer based assessment that is minimally time consuming for the assessor (compared to similar products). Drag and drop of a text editor (file extension of .doc, .txt, .rtf, .hwp, .htm, .html, all files that can be copied to mouse clipboard, etc) enables conversion into a set of questions ready for computer based assessment which the assessor can rely on.

Method

The purpose of this paper is not to introduce the whole Web Based Assessment architecture which includes hardware and software architecture with web based technology but to mention some portion of the user interface which interacts with assessors for question design and related engine.

The architecture design is separated into two parts:

1. Design of an editor with similar accomplishments to the mouse clipboard to paste text and images. Also an editor having language interchange flexibilities.
2. Text mining and ontology based architecture for parsing text to distinguish between question text, question description, images, multiple choices and symbols.

Design Html Text Editor

In order to create an online environment for the assessor who can create questions anywhere and anytime we need to provide our solution with HTML (Hypertext Markup Language) tags on a web browser. A text editor with similar functionality as common word processors could be used in our case. The reason for this limitation is because most assessors are familiar with the windows environment for creating text with Microsoft tools such as notepad, wordpad, Microsoft office tools, etc. Otherwise time needs to be spent educating assessors on new software which is not a good idea as doctors specially, have no time for learning new interface with complicated functionality. We strongly recommend use of a common design interface rather than creating a new design; something that users are familiar with.

For our purpose we tried different HTML text editors. There are many versions of HTML free text editors that we could use for our purpose without reinventing the wheel. The WYSIWYG (What You See Is What You Get) HTML Editors/Site Builders allow editing of normal text with images inside a web page somewhat like a word processor. Site Builders that allow designing online offer pre-packaged templates (and sometimes also allow a WYSIWYG interface to change elements from that template). Most popular text editors are HTML Text Editor (available at: <http://sourceforge.net/projects/richtext/>), CreaText (available at: <http://creatext.sourceforge.net/>), Cute Editor (available at: <http://cutesoft.net/>), RichText Editor (available at: richtext.sourceforge.net),

W3C eXtensible Markup Language (XML), Version 1.0. (available at: <http://www.w3.org/TR/1998/REC-xml-19980210>).

The problem with these text editors are their compatibility. Even though they are embedded into a HTML page to be online text editors, they rely on client's machine. They can load files/ save files from/to the client's computer. They are created upon scripting language and run totally on the client's machine so there is no way for the server to access the data unless client requests by sending a response to the server (e.g. either clicking a save button or refreshing page). In our case we used HTML Text Editor and in order to solve this issue, a link was opened from HTML scripting function of editor to be called into server side so that server could access the original text in rich text file (rtf) format. Figure 1 shows the features of the editor. It illustrates a portion of a radiology image. An ftp (file transfer protocol) server could solve the uploading portion by asking the user to put their images into a ftp site which is linked to the original examination paper.

Programming part of the "SaveDocument()" function (Figure 1) has been changed in the way that it can save the content of the text editor on the server and not on the client machine. Giving directions for saving content in a specific directory for further use and content management has been considered but it is not the main focus of this paper. This work can be upgraded by adding ASP (Active Server Page) functionality for server side operation.

Figure 1: Content of the SaveDocument Function

```
//SaveDocument uses the common dialog box object to display the save as dialog, then
writes a textstream object from the value of the div's innerHTML property
function SaveDocument(){
//Setting CancelError to true and using try/catch allows the user to click cancel on the save
as dialog without causing a script error
    cDialog.CancelError=true;
    try{
        var fso =
            new ActiveXObject("Scripting.FileSystemObject");
        //The path of server's machine were defined
        var f = Server.Path("../CBT/ModelFinger.rtf");
        f.write(oDiv.innerHTML);
        f.Close();
        sPersistValue=oDiv.innerHTML;}
    catch(e){
        var sCancel="true";
        return sCancel;}
    oDiv.focus();
}
```

The assessor can now drag mixed text and images from different source files into the text editor. For technical clarity we wish to mention that dragging an image brings the link it has with the original image on the client's machine and we used it to ask the user for further upload into the ftp server. Editing of existing text or typing new text is allowed using this editor and the only thing left for the assessor to do is to name the document and save it on a related directory on the server. The assessor can check the test later or wait a few seconds for the text mining algorithm to convert this document to a Computer Based Assessment ready for use.

For future work we plan to add a validation algorithm to check whether the input text is correctly converted into a Computer Base Assessment. If not it will give suggestions for conversion or await the assessor to correct the corrupted part by readjusting the text or by putting delimiters between the question, description, image and multiple questions.

Text Mining and Ontological Based Analysis Architecture

After receiving rtf text format from the client through a html text editor, a text mining and ontological assessment algorithm will separate the mixed text and image and fix them as a multiple question Computer Based Assessment.

Text Mining (Language Processing)

In order to parse one set of questions from mixed text and images we needed some defined symbols of a text in which we could track down the parsing correctly. Setting some

rules for assessors to separate portions of each description, image and multiple questions was relatively simple and we tried not to set limitations for assessors. After the first test by letting assessors create their test exams, we observed that many assessors followed similar rules and that we could set a common regulation for our text recognition algorithm as follows:

1. Question part:
 - A. A question starts with marking such as 1), 2), ..., 20) etc
 - B. Sometime a question may be set by end markings such as ?, ,, !
2. Description of a question:
 - A. Description of a question sometime marked by period followed by 'a)', 'b)', 'c)' etc.
3. Multiple choices:
 - A. Multiple choices are mostly 4 or 5 sentences that are numbered in sequence. Sometimes two multiple choices are placed in one line, having numbers between them.
4. Images or tables:
 - A. There were limited images formatted differently with text such as: jpg, gif, bmp and tif [PACS (Picture Archiving and Communication Systems were most frequently used images that were print screened by assessors)]. We observed that in general there were limited styles for each set of questions which could be repeated. Figure 2 and 3 show common questions that could be parsed without difficulty by following the above rules.

Figure 2: Text editor for assessors to edit questions and create Computer Based Assessments

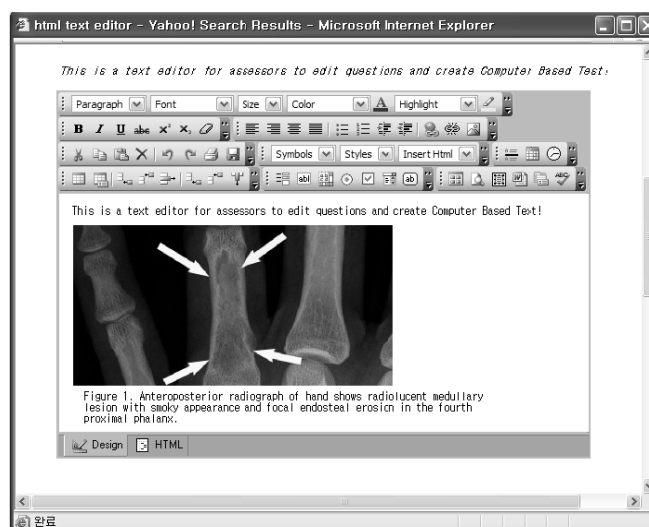


Figure 3: Example of existing PBT questions to be converted into CBT

In order to parse above mentioned rules from the mixed text and images we used XML (eXtensible Markup Language). The rules were put into were put into a XML file and loaded for parsing a set of questions. The reason for this was because for further extension of parser we needed to add ontology environment which was in XML format.

Ontological Based Analysis

Figure 4: Examples of existing PBT questions to be converted into CBT

Patient ID	Special Instructions	Allergies
Mr A Pan 1/1/39 1 Boat Street Fairland Hospital no 234567		Penicillin

Name	Dose	Times to be given
CO-AMOXICLAV	625 mg	0800
Route	Comments	1200
Oral	For 7 days	1800
Frequency		2200
QDS		

Among the set of questions that the assessor saved on the server, some did not exactly follow the rules mentioned above. Figure 4 illustrates one of the examples that we had difficulty parsing using simple rule based text mining. As shown in the figure, symbols (e.g.number 1. and 2. in the figure) marked in description and multiple questions confused the parser by recognizing them as regulations mentioned above.

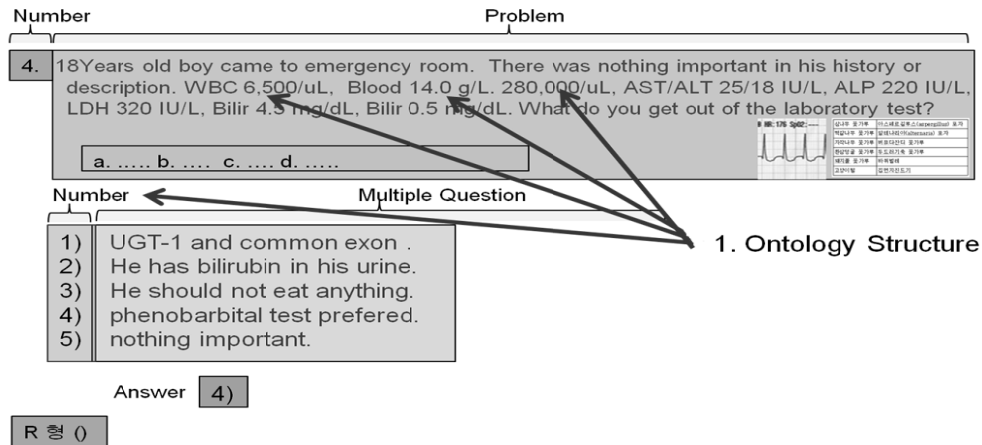
Problems with repetition of the same symbol used for marking different meanings was among the most frequent mistakes that computer could not distinguish. In order to solve this problem we had to recognize the pattern by creating related ontology of the symbols and categorize them in the way that parser could distinguish their differences. We tried Protégé development tool. The Protégé environment has been developed at Stanford University over the past 16 years (Protégé project, Stanford University, <http://protege.stanford.edu>). It supports the modeling of ontologies and use of ontologies to guide acquisition of content knowledge from subject-matter experts. Moreover it allows developers to easily “plug-in” components to add new functionalities to the Protégé tool. Protégé is being actively used by hundreds of users world-wide in many knowledge domains.

Figure 5 illustrates the classes made by the Protégé tool. We created four classes as we had four major portions in one set of questions such as: Description, Image, Question and Multiple questions. We added another class named Marks that held Numbers, Characters and Symbols. Numbers were a set of individuals from 0 to 9. Combinations of them could lead to all available numbers. Characters were a set of ‘a’, ‘b’, ‘c’ (individuals) for describing explanations in sequence. They could have been included in the Multiple Questions as well so we added them to be individuals related to Description class “same as” Multiple question class.

Results and Conclusions

In one example of converting a set of questions made by an assessor, we tried our solution to see the performance results. Among 380 questions set in the area of Pathology designed by assessors at different time schedules, 65% of question sets were parsed successfully using the methodology described above, without problems. The rest (35%) failed.

Figure 5: Ontological Structure of the raw material to be parsed.



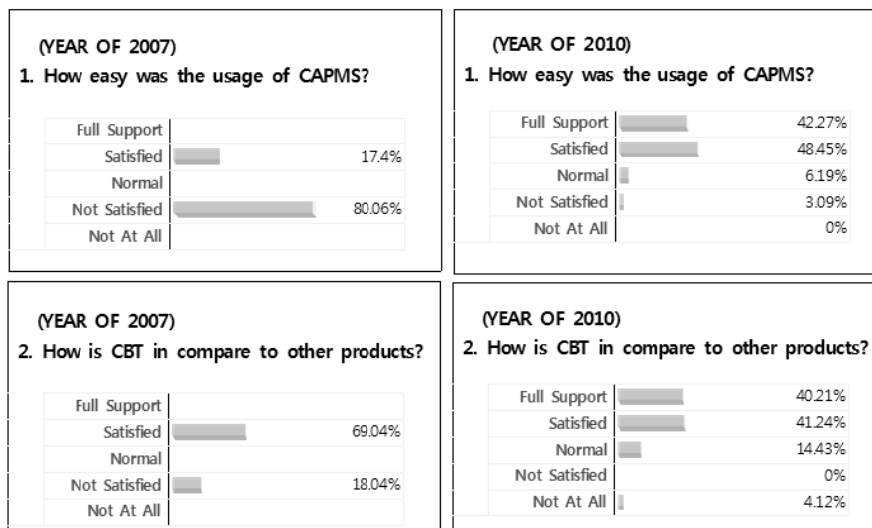
Percentage analysis of the failures are as follows: 77% were parsed without difficulty by ontology method described above, but the editor could not handle the mixture mode of answers to be described just after the question, or answers to be described after multiple questions. In that case we should have described answers just after the question and before multiple questions so the methodology could work without problem. This issue can be considered solved if we could add one more function to parse the exam once more and before starting to handle methodology above to replace and unify the exam answers to appear just after each question.

The remaining 23% were unresolved and left challenging for us to be considered in future work. To solve this issue we marked four portions of a question set manually (e.g. Question, Description, Image and Multiple

Question) and created test pages for assessor that they could follow.

The total time for converting 380 question sets was about 162 seconds on a dual core computer (the performance will be enhanced by using a server which is not considered for now). Finally, performance of our test result was faster than any other web based application solution. With the result of this research we find ourselves a little closer to a user friendly environment and full automation of web based assessment. The achievement of work described in this paper was to reduce the amount of time the assessor spent dealing with software to create questions for computer based tests, also reducing usage difficulty level of software to be more convenient for assessors which is the major issue of software development these days.

Figure 6: Level of software convenience illustrated as satisfactory in users point of view. Question answered by different assessors in different timing schedule (2007-2010)



In real teaching, we asked 10 assessors for usage feedback of the CAPMS (Computer Assessment of Pusan Medical School) tool. Figure 6 illustrates the improvement and level of convenience of the current methodology. During the year 2007, questions were answered by different assessors and the result was unsatisfactory. The methodology explained in this paper shows the level of convenience of usage if more user friendly issues take part in implementing such software.

References

- Appel, J., Freidman, E., Fazio, S., Kimmel, J. & Whelan A. (2002) Educational assessment guidelines: a clerkship directors in internal medicine commentary, *American Journal of Medicine*, 113, pp.172-179.
- Bugbee, A.C. (1996) The equivalence of paper-and-pencil and computer-based testing, *Journal of Research on Computing in Education*, 28, 3, pp. 282-299.
- Choi, S.W. & Tinkler, T. (2002) Evaluating comparability of paper and computer-based assessment in a K-12 setting, paper presented at the Annual Meeting of the National Council on Measurement in Education, New Orleans, LA.
- Plichart, P., Jadoul, R., Vandenabeele, L., & Latour, T. (2004) TAO, a collaborative distributed computer-based assessment framework built on semantic web standards, paper presented at the International Conference on Advances in Intelligent Systems – Theory and Applications AISTA, Luxembourg.
- Pitoniak, M. (2002) Automatic item generation methodology in theory and practice (Center for Educational Assessment Research Report No. 444), University of Massachusetts, School of Education, Amherst, MA.
- Revuelta, J., & Ponsoda, V. (1998) A comparison of item exposure control methods in computerized adaptive testing, *Journal of Educational Measurement*, 35, pp. 311-327.
- Thelwall, M. (2000) Computer-based assessment: a versatile educational tool, *Computers and Education*, 34, pp.37-49.
- Seale, J. (2002) Using CAA to support student learning [online] LTSN Generic Centre, available at: www.ltsn.ac.uk/application.asp?app=resources.asp&process=full_record§ion=generic&id=38 [accessed 18 Jul 2004].
- Yi, Q. & Chang, H. (2001) A-Stratified computerized adaptive testing with content blocking Paper presented at the Annual Meeting of the Psychometric Society, King of Prussia, PA.