Distance Education Assessment Infrastructure and Process Design Based on International Standard 23988

Steven C. Shaffer, DEd.
The Pennsylvania State University
scs12@psu.edu

Abstract

Assessment is an important part of distance education (DE). As class sizes get larger and workloads increase, the IT infrastructure and processes used for DE assessments become more of an issue. Using the BS ISO/IEC 23988:2007 Standard for the use of technology in the delivery of assessments as a guide, this paper describes a rational approach to an information technology infrastructure and process which helps to answer questions regarding the quality and integrity of DE based assessments.This paper remains agnostic with regard to the validity of types of assessments used for various purposes, and deals only with the issues of IT delivery and the management of assessments. The information presented can be used as a basis for in-house written software development or to compare and contrast various off-the-shelf options.

Introduction

Assessment is a particularly thorny aspect of distance education (DE) course delivery; various researchers and practitioners hold strong beliefs with regard to the validity, reliability and fairness of various methods of assessment. That being said, most for-credit distance education courses contain some form of assessment, and the organization and management of these assessments needs to be considered. BS ISO/IEC Standard 23988:2007 contains guidelines regarding the appropriate use and management of information technology (IT) in assessing students. Briefly, the guiding principles of the standard are: (1) assessment reliability and validity should not be reduced due to the IT delivery mechanism; (2) the assessment should be fair to all students (including making adjustments for students with disabilities); (3) health, safety and confidentiality should be maintained; and (4) the security of the data, audit ability and backup/recovery ability should be maintained (BS ISO/IEC, 2007).

Much of the information contained in BS ISO/IEC Standard 23988:2007 would be familiar to any quality assurance (QA) professional, however it does serve as a good structure for QA-checking one’s infrastructure and procedures. With that in mind, it should also be possible to design a comprehensive IT process and infrastructure which fulfills most if not all of the proposed procedures. The following discussion is the result of attempting to do just that. In the discussion below, numbers in square brackets represent the section number in the standard that refers to each design decision; this is for reference for the reader and to allow a “trace back” to the standard if desired. For a full understanding, the reader should obtain and read the BS ISO/IEC Standard 23988:2007 itself.

New User Registration and Validation

Figure 1 shows the process of receiving a new user registration and validating the student for later assessment. This step is necessary for later verification of the identity of the student who is taking the assessment [15.3.1]. One path shows the student registering with the institution (usually this would be course registration); after doing so, the institution sends the registration information to the DE administration team. This is a standard process; the only thing that needs to be mentioned here is that this process must employ state-of-the-art security methods, including using secure socket layers for transport and appropriate encryption and password protection for storage [6.4.2.1]; no emailing the registration list! (All data transfers discussed in this article must also employ these same protections.)

The other process path through Figure 1 begins with the student transferring an image of an official identification document to the DE administration team. (If the institution has an electronic image of the student on file, then this step does not have to be performed.) At least three mechanisms can be employed here; (1) the student can photocopy the identification and send the copy through the postal service; (2) the student can take a photograph of the identification document and send it electronically; or (3) the student can scan the document and send it electronically. In cases (2) or (3) the administration should supply a secure application for transmitting and storing the file; again, email is not secure enough. In addition, having an application to upload the file also means that it can be stored in a database or in another way such that it can be automatically accessed when needed [10.2.1]. If an outside test security group is to be used, information needs to be securely transferred to that organization as well. During the setup of the student account, entry should be made to indicate special accommodations for students with disabilities; for example, a database field could be populated with a percentage of extra time allowed [7.4.3]. The student is then supplied with a username and password for access to the assessment system [6.4.2.1].

The final step in the registration process is to have the student log in and take a sample assessment which does not “count” except to verify that the student has the appropriate access and equipment needed [13.3.1] to execute the later assessment(s); ideally the software will test for appropriateness and display any incompatibilities. Doing this covers section [6.2.2] which is concerned with the student having the appropriate hardware and software needed to run the assessment(s). Additionally, this step should present at least one example of every type of assessment item that will be encountered in the real assessment(s) [6.5.2, 12.2.1]. The instructional systems staff will need to review the results of the assessment, along with any comments from the student, both to validate that the student is able to take the assessment(s) and also to inform the development staff of any enhancement suggestions.

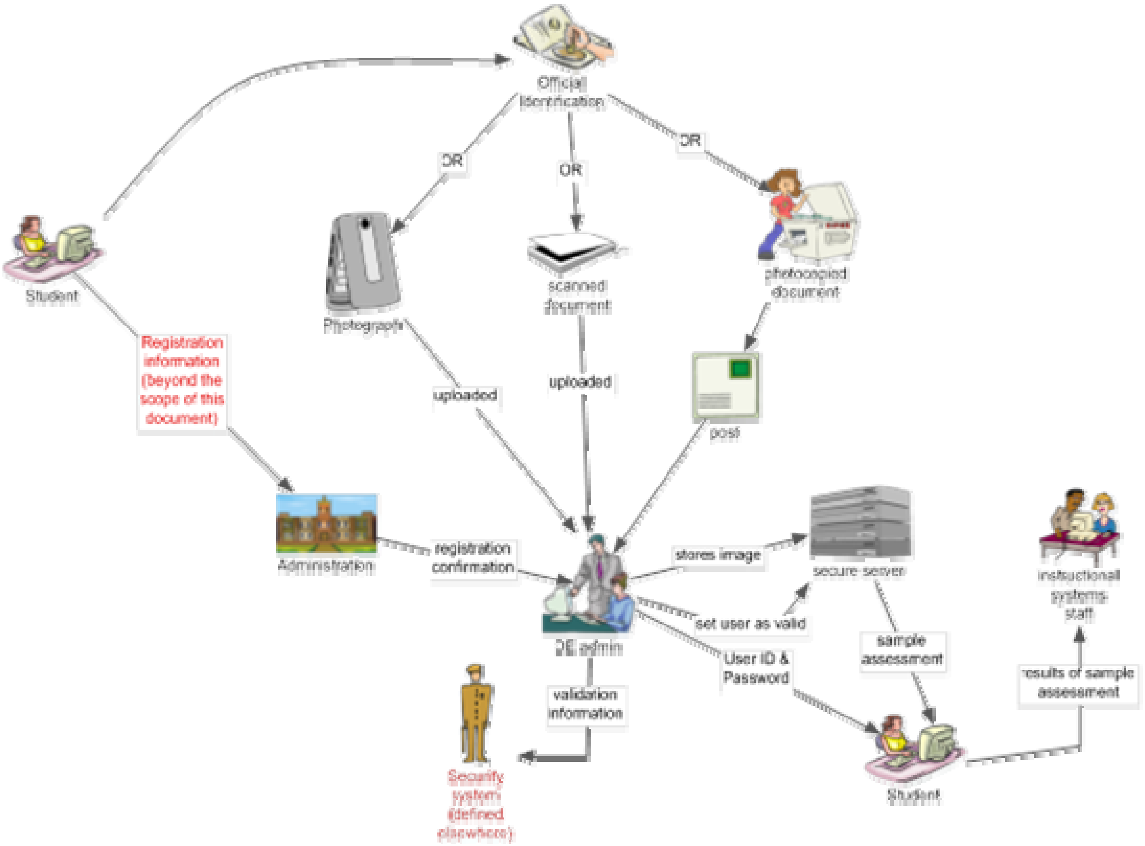

Figure 1. New user registration and validation

Test Setup and Quality Assurance

Figure 2 shows the steps involved in developing and validating an assessment.This process starts with a tight, iterative process between the faculty member and the curriculum committee to develop an appropriate assessment. This should include the types of items, the number of items, time limits, etc. [5.1.2]. Once the assessment is agreed upon, it is forwarded to the instructional systems staff for implementation [5.1.3, 6.5.1.2]; again, this should be a secure transfer and not done via email. Utilizing the appropriate assessment software, the instructional systems staff will implement the assessment for electronic delivery. During this step, the instructional systems staff may need to interact with the faculty/curriculum committee with regard to specific ways to translate the assessment into an appropriate electronic form.

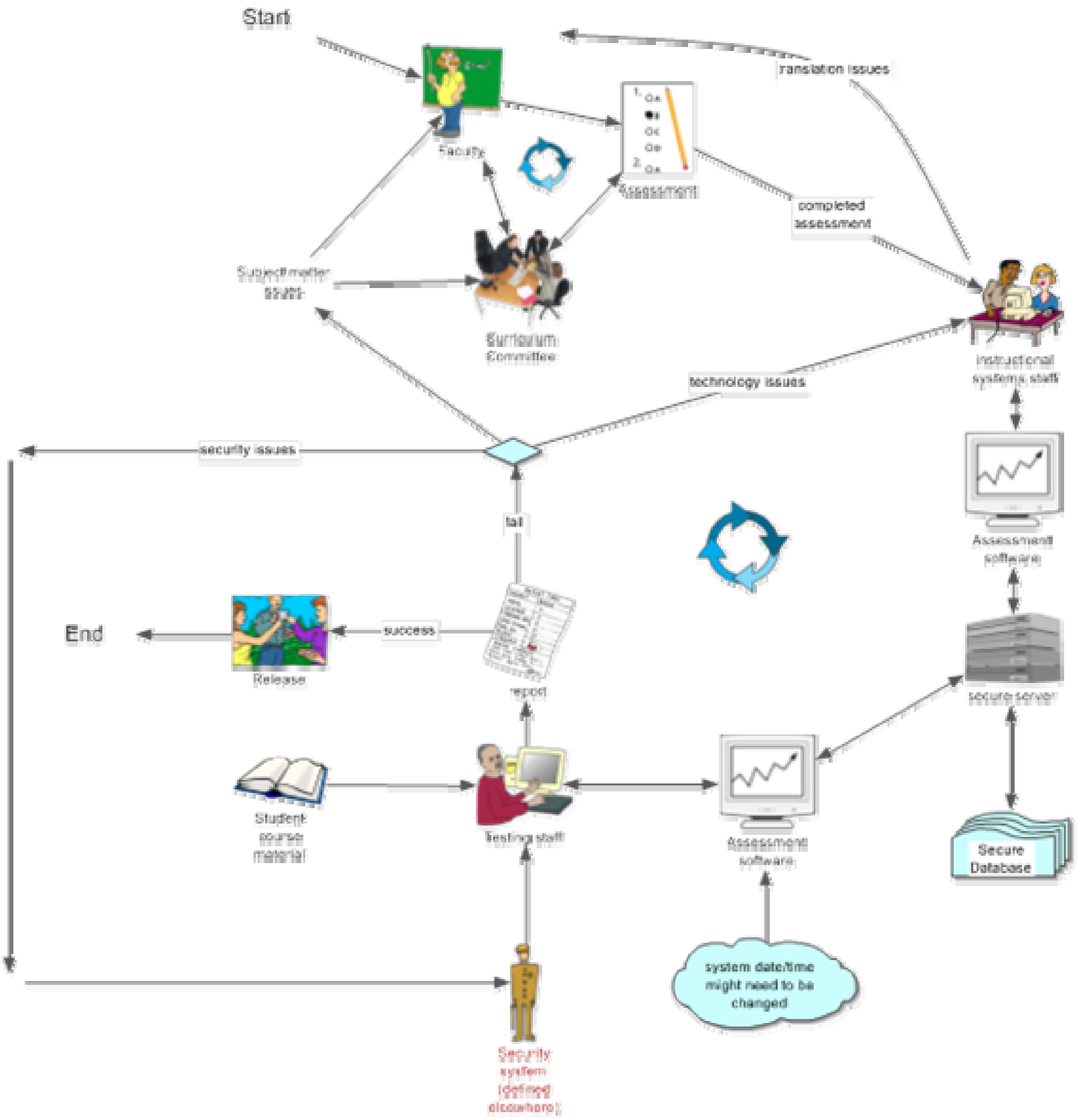

Figure 2. Assessment creation and quality assurance

Once the assessment is implemented within the assessment software, the testing staff will log in with a student-privilege-only account and test the assessment for several of the following qualities [6.5.1.2]. Technical issues involving how the assessment is set up in the assessment software are routed back to the instructional systems staff; pedagogical or subject-matter issues will be sent back to the instructor/curriculum committee [8.1.3.5]; security issues would be forwarded to the test security group [5.1.3]. In order for this step to be completely valid, the tester must be a non-expert whose only knowledge of the subject comes from having accessed the course materials [7.5.1]. In addition, having the assessment performed by a person needing assistive technologies would be beneficial. If all goes well, the assessment is released and can be made available for the student.

Performing the Assessment

Figure 3 explicates the process of performing the assessment online. It starts with the student logging in using a user name and password [6.4.2.1], and possibly a code specific to the assessment to be taken.The student is presented with choices of assessments, or the software might just direct the student to the one and only option. Next the student is presented with a set of instructions for the assessment, including the expected time to completion, use of course materials, topics covered, scoring [8.1.4, 12.1.1] and the policy on breaks [7.2] and appeals [11.4]. Note that all of these instructions may have been available before the assessment as well. In addition, if any informed consent is required regarding the use of surveillance during the assessment or for data collection, it should be done at this time [6.4.3.2, 12.4]. The assessment timer is not activated during this setup process [6.3]. The student next has the option to start the assessment or cancel the process. If the student cancels the process, no penalties should be assessed unless the assessment was to be done at a particular date and time.

Next, the security procedures (if any) begin. According to [15.3.1], it is incumbent upon the organization giving the assessment to verify the identity of the person taking it.There are several possible security approaches. The traditional approach is to have a proctor (a.k.a. invigilator) present in the room while the student is taking the assessment. As DE becomes more common, this approach to security begins to be unworkable. One issue with this approach is that it disadvantages students who do not live near testing centers or other suitable venues, as well as disabled student who would find it difficult to get to a testing center. Two other issues are cost and scheduling, both of which can severely interfere with the *zeitgeist* of a distance education course. A currently popular approach is to use video monitoring software to have the proctor remotely monitor the student.This approach has the advantage of not requiring the student to be physically near the proctor, but still has the disadvantage of requiring that a real-time monitor be available. It also requires a constant and stable internet connection. In the design proposed below, images of the computer screen and the student are captured at random intervals and stored into a packet of data to be uploaded when the student completes the assessment (see more on this below). This type of surveillance will require student permission [6.4.3.2].

Once the student has agreed to begin the assessment, a secure socket layer connection is used to download an encrypted data package onto the student's computer; a hash value is used to ensure that the assessment has been properly received [9.2.2, 15.2]. The assessment is stored locally in an encrypted format [6.4.2.3, 13.5.2]. This differs from common practice, which is to use an interactive browser-based assessment which requires continuous internet access. The different approach advocated here is based on standard sections [6.2.1] and [6.2.4.2] which states that "measures should be taken to ensure that candidates are not disadvantaged by slow connections" and which specifically suggests "downloading the whole assessment onto a local network or hard disk just before the assessment begins" [6.2.4.2]. Having the software resident on the student's computer throughout the assessment helps to guarantee that the student is not disadvantaged due to a slow or intermittent internet connection. Pragmatically, it also reduces the opportunity for the student to *claim* that this has happened. Once the test package has been received and verified, the test timer can begin [6.3]. If the assessment is not timed, this step can be skipped. The main assessment software now executes, treated as a black box here, but described in detail below.

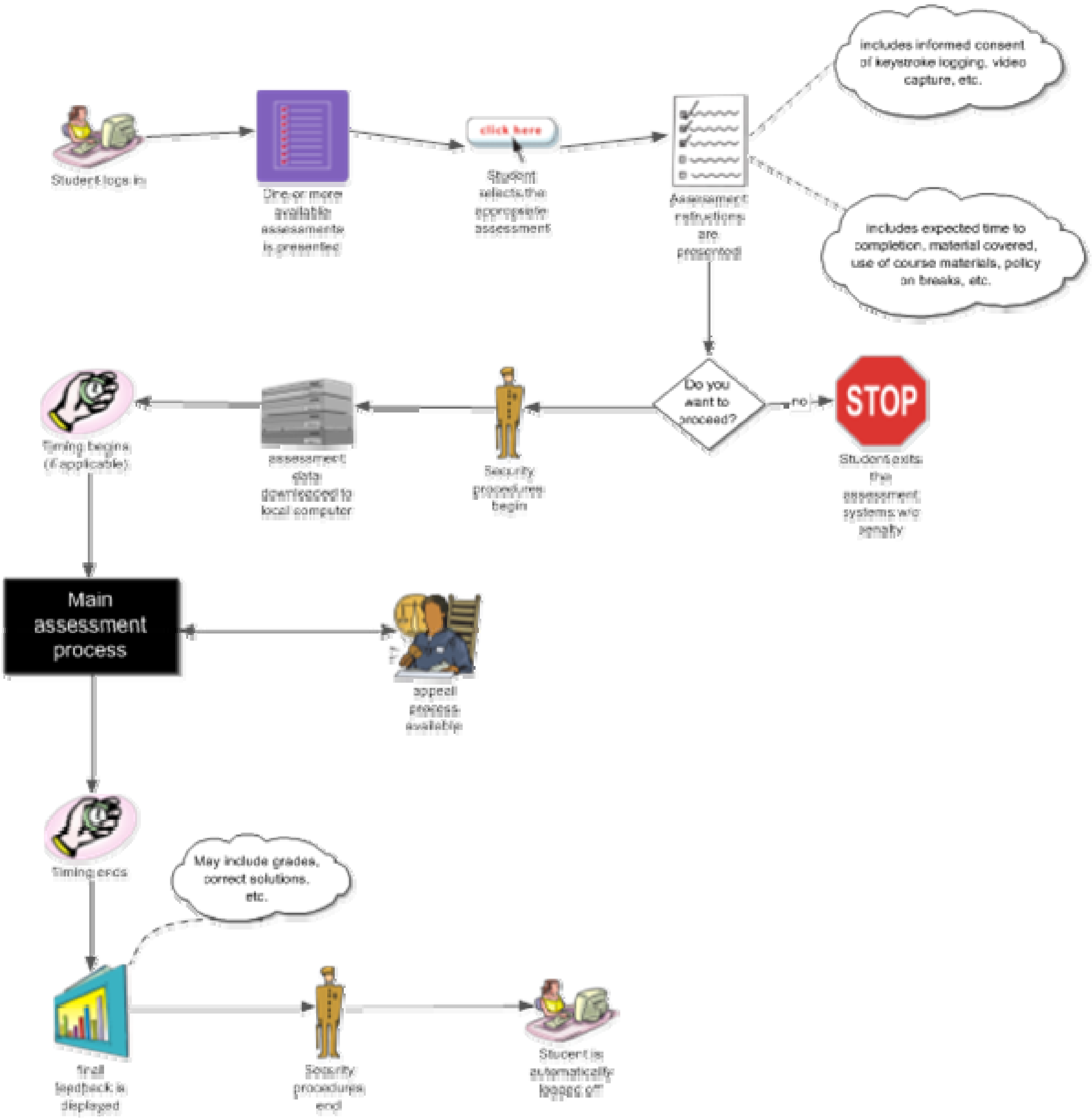
Student logs in
One or more available assessments is presented
click here
Student selects the appropriate assessment
Assessment instructions are presented
includes informed consent of keystroke logging, video capture, etc.
includes expected time to completion, material covered, use of course materials, policy on breaks, etc.
Do you want to proceed?
no
STOP
Student exits the assessment systems w/o penalty
Security procedures begin
assessment data downloaded to local computer
Timing begins (if applicable)
Main assessment process
appeal process available
Timing ends
May include grades, correct solutions, etc.
final feedback is displayed
Security procedures end
Student is automatically logged off

Figure 3. Performing the assessment

During the assessment the student can press a “hot key” to initiate an appeal. Once an appeal is requested, the timer stops running and access to the assessment is disabled. If live support is available, the software will attempt to connect the student with it. If not, the process will allow the student to register a complaint and then ask if the student feels reasonably able to continue. If the answer is yes, then the timer is re-started and the assessment continues as normal. If the answer is no, then the test data is uploaded to the server and the assessment shuts down for later manual analysis and rectification.

Final results are displayed (if appropriate) and the security system is shut down (for example, if a live proctor is used this is the end of his or her duties). If scores are given, they should be listed as provisional until the results can be validated [8.3.3.1]. At this point the student is automatically logged off of the assessment system.

Requirements of the Assessment Software

There are a number of aspects of the standard which either directly state or imply specific attributes of appropriate assessment software. However, many of these statements are in the form of conditionals; e.g., if your software does x, be sure it handles y. Therefore, this section of this paper is less straight-forward than the other sections because of the degrees of freedom in the design. Thus, a sample software design is proposed, along with justifications from the standard; we will call it program P. Although generality is a goal with this example, this particular design may or may not apply to any specific institution's organizational or pedagogical situation. Please refer to Figure 4.

Program P is written in Java for the following reasons: Java (1) is cross-platform [6.2.1], (2) can be run on the desktop (see previous discussion on avoiding latency problems); (3) has built-in support for many/most AV formats [6.5.1, 6.5.2]; (4) can support specialized fonts and symbols for mathematics and foreign languages via built-in UNICODE support [6.2.5]; (5) has assistive technology libraries available, and (6) has good support for standardized file formats such as CSV and XML, which are important when transferring data in and out of the system [6.1.1, 10.1].

The standard suggests that the assessment program should disallow access to outside programs when it is running [16.1.2], or perhaps not re-start if another program is accessed during the assessment [6.4.3.1]. This kind of restriction can be accomplished, but requires operating-system specific setup with administrator privileges, making doing so contraindicated for a DE application for the following reasons. First, being operating-system specific runs counter to [6.2.1] wherein it is usually preferable for the application to run on multiple platforms. Second, remote students who may be using shared computers or computers managed by their workplace may not have sufficient privileges to run such software, nor would their IT departments take kindly to it. "Pretending" to take full control of the computer by displaying full screen, removing menus, and disabling "hot key" buttons will not deter the sophisticated computer user, and it is one of the basic principals of the standard that students who are not tech-savvy should not be disadvantaged by the assessment system [4]. Since the goal of [6.4.3.1, 16.1.3] is to limit a student's access to "helper" software which would essentially allow the student to cheat, an alternative to the suggestion of "locking down" the application is the surveillance cycle proposed above; by taking screen shots either very often or randomly, it should be possible to catch students who are cheating [6.4.3.2]. At the very least, announcing that this is what the software will be doing should ameliorate cheating by virtue of the increased chance of being caught. Of course, if the student is allowed access to certain digital assistants, these will not be counted as cheating. Care should be taken to be sure to capture screen images of all running video monitors, in case the student is using multiple monitors.

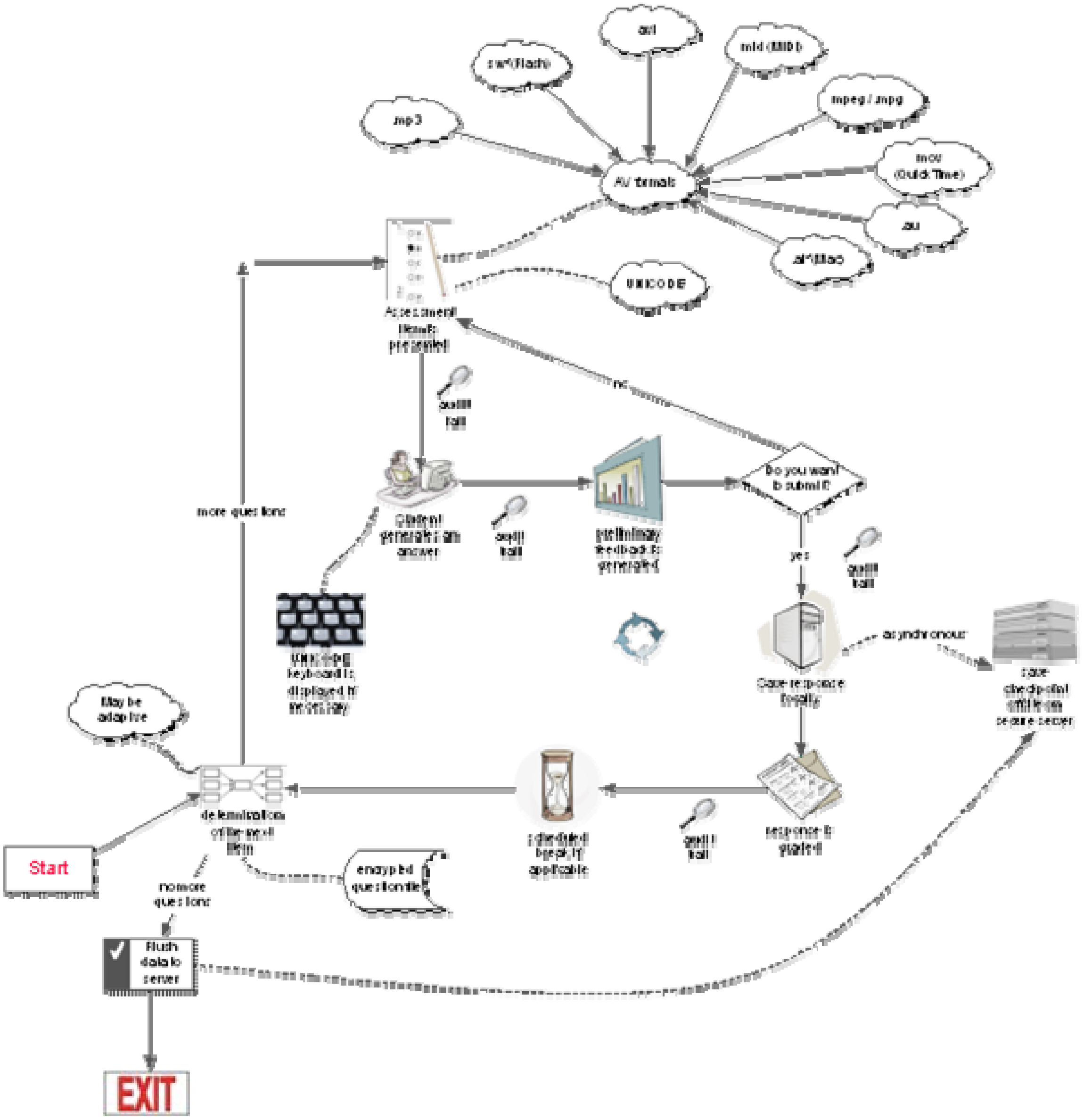

Figure 4. Sample assessment software design

With regard to ensuring that the student taking the assessment is in fact the student who is registered for the course [15.3.1], the proposed design includes taking randomized pictures from a student's web camera. There are both technical and privacy issues here. From a technological standpoint, this means that each student's computer will need to include a web camera; based on the price points of laptops with built in cameras and the popularity of such applications as Skype and Google+, requiring that a student have a web camera available is becoming less and less of an issue. From a privacy standpoint, it is important that the software require informed consent before starting the image processing part of the program. It's also important that every safeguard be established to ensure that the image capturing process is turned off when the assessment software is finished. Once established and running, the software will capture images of the student working on the assessment. This approach to capturing video images of the screen and the student also helps to capture technical failures or anomalies [17.4].

With both the screen and student image capture in place, administrative personnel and faculty can check the images while reviewing the results of the exam. In any but the simplest of cases (e.g. low stakes/multiple choice assessments) there should always be a visual review of the assessment results. Replaying the assessment script, including displaying the images, should allow a reviewer to check for anomalous results and behavior within a very short amount of time. The review software also needs to allow the reviewer to freeze and reverse the display. In P, these images are embedded with the rest of the audit trail so that the entire student session can be “played back” after the assessment has been submitted.This approach to proctoring is not foolproof and thus it is not suggested that it be used for very high stakes testing (e.g., medical licensing); however, being able to circumvent this surveillance requires a great deal of sophistication and should be sufficiently secure for most coursework. In practice, it may be more secure than a typical face-to-face final exam involving hundreds of students in an auditorium. An additional advantage to this approach is that it affords the opportunity for an instructional systems staff or faculty member to check the results of automated scoring [8.1.2, 8.1.3.4] and review the assessment for future changes and optimizations [8.1.3.5].

In a similar vein, the standard suggests that the software should disable printing, copying, sending or saving files which might be used later to cheat [6.4.4]. The purpose of this recommendation is to maintain the security of the assessment items [9.2.1]. This is somewhat workable in that the software itself can prevent reading from and writing to the clipboard, printing, etc. However, if the student is willing to take a screen shot (or even use a camera or write down items), then the concept of full security of the assessment items is unrealistic. An alternative approach, if feasible, is to generate assessment items on the fly. This can be done with certain types of math, science, or computer programming problems but is unlikely for less computational domains. Essay answers might be handled post-hoc through an automated interface into a plagiarism detection program. Multiple choice exams can be somewhat protected by using large question banks, so that a copy of any one student’s exam will have little value for any other one student. In the end, however, once any student has taken an assessment outside of a proctored environment, the questions must be considered compromised.

Design of the Sample Assessment Software P

Beginning with the overall process described previously, the assessment itself proceeds as follows. The student is presented with an assessment item. Upon answering or solving the item, the student enters a response. At this point the assessment system may reply with some preliminary information. For example, if the student was writing a computer program, this step might return the results of a compilation; if the student is generating a graph, the system might present the graph for verification. This stage is dependent upon the nature of the assessment and the assessment software.The student is asked whether or not the answer should be submitted; if so, then the software stores the answer locally (encrypted and secured) [7.1.4, 13.5.2]. If there is any feedback at this point, it is presented to the student; for example, the software might report “3 of 8 questions correct” [7.2]. Extra care must be used when the software will automatically score the items. For example, in open-ended response items, issues such as upper/lower case, misspellings and punctuation must be taken into consideration [8.1.3.1].

The program should present a consistent interface to the student so that confusion does not effect the assessment score [7.1.1, 7.3.1]. If the student is allowed to leave items unanswered, including being able to return to these items later, the mechanism for doing so should be obvious [7.1.2]. Pragmatically, this requirement argues for an interface which only presents one item at a time and for which backtracking is not allowed; this approach also helps with break processing (see below) and works well with adaptive testing. Fail-safe measures such as *are you sure?* prompts should be included to avoid accidentally submitting an answer [7.1.3] or accidentally exiting the assessment [7.1.5, 7.3.2.1].

During this process the software should store checkpoints which will record the student’s progress to that point so that the process can be re-started if there is a power failure or other technical issue [6.4.2.2, 10.2.2, 17.2, 17.3]. When the student submits a solution, all of the data, including the surveillance data, can be uploaded securely to the server via a process that runs asynchronously in the background [10.2.1, 10.3.1, 13.5.2]; if the sever is currently unavailable, the process can periodically “wake up” and check for server availability. Using this method, the student does not have to wait for a slow or non-existent connection [5.1.5].

Next, the software will determine if a break is appropriate [6.4.5]. This might be done by simply asking the student, or it might be programmed into the software. If a break is commenced, the assessment timer should be paused; when the break is over, the assessment timer will restart [6.3].

Next, or after a break if there is one, the software must determine what the assessment item is. This might be done by simply going to the next item, selecting a randomized item or using an adaptive testing algorithm.

Once the student has answered all of the items, the data are saved to the server. If this process is interrupted or if the server is not found, the process is re-tried. After the third try the student is informed that the test data has not been loaded and to try again later; instructions on how to do that are displayed. The data are securely stored on the student's computer and should be encrypted and tamper-proof [6.4.1].

Adjustments for Students with Disabilities

For ethical as well as legal reasons, it is important to ensure, to the extent possible, that the assessment system does not unnecessarily disadvantage students with disabilities [5.1.4, 7.4.1]. An important adjunct to this, though, is that the system should also not give an unnecessary advantage to the disabled student via the results of using assistive technology [5.1.4]. For example, a chart or diagram might be fitted with an ALT tag that includes an explanation that leads the student toward a certain conclusion which is not normally accessed by the non-disabled student. Color and font selection should also be available for the visually impaired [7.3.1]. Keep in mind that the use of certain assistive technologies might slow down the student's progress through the assessment, and accommodations for extra time may be needed. If extra time is needed, this can be added to the student's record when s/he registers; the software should then automatically allow more time based on this entry.

Conclusion

BS ISO/IEC Standard 23988:2007 very clearly states that conformance to the standard does not confer immunity from legal obligations. In addition, it clearly states that it does noSt contain enough detail to be considered a contract. Instead, the document is meant to be used as a suggestive list of important aspects to consider when performing student assessments via information technology. Likewise, this paper does not purport to include every aspect which may be required in any particular situation; instead, it presents a template for developing solutions specific to a particular institution. In addition, this paper does not act as a substitute for the standard itself, which is very detailed on a number of topics not discussed here.

Acknowledgements

The diagrams in this paper were generated using Inspiration by Inspiration Software Inc. The icons/images come from that software.

Thanks to Thomas Iwinski of Penn State University for reviewing and commenting on this manuscript.

References

BS ISO/IEC 23988:2007: Information technology – A code of practice for the use of information technology (IT) in the delivery of assessments. British Standards Institution, London, United Kingdom.

Back to the Online Journal of Distance Learning Administration Contents