Honours Project Report

ScriptView

Victor Soudien

Supervised By: Gary Stewart

Mark Table

# Introduction

The management of test scripts at tertiary institutions involves collection and distribution to markers, evaluation and redistribution to students. The current methods in place rely on the physical test scripts, require a considerable amount of time and coordination between multiple individuals. These methods increase the waiting time for students who ideally need to get feedback as soon as possible in order to improve for future assessments. Current test management also lacks the collection of any data about the tests aside from the student’s final mark. This leads to a lack of historic data for educators to reference when creating new tests.

Needs more.

# Background

This section presents the relevant findings from previous and current work that either directly or indirectly influenced the design and development of the system.

## The current procedures

Observation

## Functionally similar software

Despite the increasing prevalence of technology in education, the marking of paper-based tests has remained largely unchanged. There has however been significant work done on trying to digitize the marking process which shows some promising results.

One of the most widely recognized techniques is that of Optical Mark Recognition (OMR) which utilizes optical equipment to interpret marks on a paper [1]. These marks are usually indicated in a grid format with cells corresponding to letters or digits. Commercial products which implement OMR are available and have had limited success. One of the key constraints of OMR is that it limits the examiner in terms of what type of questions they can ask. This is because OMR is only suited to multiple choice which implies that examiners can only ask questions up to the applying level in Bloom’s taxonomy.

An alternative to this would be to use a system that could analyze free form answers. This kind of technique has been explored, especially in terms of essay marking.

Researchers such as Christie [2] have found success applying automatic marking to the marking of essays. The researcher describes a technique used to mark both style and content which requires acceptable style metrics to be set up before marking as well as manually marking a certain amount of scripts to calibrate the algorithm. Such a system would allow for questions from all levels of Bloom’s taxonomy and would therefore be more widely applicable.

Pulman [3] did similar work in that he attempted to use machine learning techniques to automatically mark short free form answers. Although he draws no conclusions on how easy his method is to customize to different questions, there is a problem which is identifiable in both his and Christie’s approaches. That problem, is training data. A system that is more generally applicable to a wide range of test is preferred since time does not have to be spent on customizing it for each new type of test.

There are also certain issues which could arise which Thomas [4] expanded upon in his work. Thomas performed an experiment in which he compared the results of a fully automated marking system and a manual marking system. The tests intended for automatic marking were taken on a computer and the other tests were done on paper. The researcher noted that certain problems could arise during the taking of the test that could only be solved effectively by human markers.

The most problematic of these being that of ambiguity, either in the student’s answer or the examiner’s question. Students may also not be able to express themselves clearly, especially if they are taking the test in a language other than their first language. Both of these problems could be overcome by manual markers who used their knowledge of the domain and discussion with fellow markers to assign appropriate marks.

## Forms of assessment

An advantage of electronically processing tests is that it creates the opportunity for the collection of large amounts of data about students’ progress. This data can then be analyzed to predict learning patterns and activities that might indicate a risk of failure [5]. As one of the main goals of the system is to allow for the recording of such data, it was essential to understand what kind of information an educator may look for and how it would be used.

It was found that tests are often divided into two distinct categories namely formative and summative assessment. Formative assessment is used during the course of a particular section such that the results can be used to inform how the student or educator should progress [6]. Summative assessment on the other hand is used at the end of a section to gauge the extent of the student’s learning for grading, certification or the evaluation of the effectiveness of the teaching method [6].

The terms formative and summative assessment however, do not describe the format of the tests but rather their function. As a result, a range of question types can be used in both. A classification of these types based on their cognitive complexity was developed in 1956 by B. Bloom and revised (see Figure 1) in 2001 by L. Anderson [7]. The cognitive level of complexity increases as one ascends from remembering to creating. The taxonomy provides a framework which educators can use to ensure that they are testing all aspects of a student’s understanding as well as allowing them to create tests which assess the most appropriate levels for the given situation.



Figure 1: Original and Revised Bloom's Taxonomy [2]

Aboulsoud [8] claims that formative assessment is more valuable to students since it provides them with feedback which they can use to improve their future work thus increasing their chances of academic success. He suggests that for formative assessment to have this effect, it should be followed by immediate feedback which clearly indicates areas in which the students should improve while acknowledging their effort to achieve the objectives of the assessment.

This implies that any system developed to aid the marking process should allow for rapid feedback with the characteristics outlined by Adoulsoud as well as allow for the marking of questions at various levels of Bloom’s taxonomy.

## Converting physical tests to digital format

The biggest bottle neck in the entire back-end process appears to be the scanning of test scripts so that they can be processed and stored electronically. This was noted by Doctor H. Suleman (personal communication, 24 April 2014) of The University of Cape Town who has implemented a system which involves the scanning of marked test scripts to automate the process of capturing marks on the university’s learning management system known as Vula. The system he developed only requires that the cover page of the test be processed since it is the one which contains the marks and student information. However, all pages are scanned since they will be emailed to the student as feedback. Due to this, the quality of the scanned document needed to be high enough so that the student would still be able to read the feedback written by the marker and that the image processing algorithms could successfully detect the student number on the cover page which was indicated by shading certain pre-defined areas.

To accomplish this the documents were scanned at 300dpi but it was decided that the scanning was too slow and thus proved infeasible. To improve the speed of scanning, the resolution was changed to 200dpi and the documents were scanned in black and white. This sufficiently reduced the scanning time while maintaining the readability of the document.

A problem that was not addressed by Dr. Suleman during our interview, was the size of the scanned documents. This was however addressed by Doctor J. Tangkuampien (personal communication, 1 May 2014), who also scanned documents at 300dpi and had to reduce it to 200dpi.He however, reduced the resolution due to concerns about the file size. When scanning at 300dpi the file size was approximately 4MB and decreasing the resolution halved this size. Another difference between his scanning solution and that used by Dr. Suleman is that he scanned the documents in colour. While scanning the documents in black and white would have further reduced the file size, Dr. Tangkuampien noticed that doing so interfered with the visibility of the text if students used highlighters. This is because when scanning in black and white, very light highlighters such as yellow do not show and dark ones obscure the text.

Dr. Suleman used a combination of OMR and manual name entry to name scanned files whereas Dr. Tangkuampien used a fully manual approach. With the first approach a special cover page needed to be created whereas with the second the existing test format could be used.

S. Chetty (personal communication, 5 May 2014) described a method which uses optical character recognition (OCR) to name files. This method required that a certain area of the script be reserved for the student number, which eliminated the need for a dedicated cover page as the area for the student number could be included on the same page as the first question of the test. This is essential as he suggests that a large part of the process will be a compromise between the existing process and the new electronic solution. By eliminating the creation of a cover page the overall process can also remain as simple as possible.

Minimizing file size while maintaining the readability of the final document is essential to the final system as limited storage is available; scripts for multiple tests need to be stores and image processing needs to be performed on the documents.

## Software interaction

In order to make the best use of the tablet and stylus interface, it was necessary to understand what users have found intuitive in the past.

Alisi [9] discusses the concept of natural interaction during the description of systems implemented to improve the experience of museum visitors. The researcher suggests that natural interaction can “reduce the gap between computing and ordinary physical things” however it would require that the interfaces differ from traditional human-computer interaction such as the use of menus and icons.

The Point At system described allows the user to point at a character in a picture in which they are interested in and the system will provide them with more information on that character. Since this a natural action and similar to how the user would traditionally ask a tour guide for information, it allows the technology to become a transparent medium instead of overwhelming the experience.

This point is further expanded upon by Malizia [10] who believes that a natural interface, especially gesture based ones, should allow users to interact with the software using the same gestures they would use with the actual physical object. The researcher states that users should not have to learn an arbitrary set of gestures to use the software but instead the software should allow gestures which take the user’s habits, background and cultural aspects into consideration. The researcher recognizes that this goal might not be achievable but knowledge of these aspects means that we can design interfaces that are as natural as possible by carefully considering what will be intuitive for the target audience.

In order to provide the students with useful feedback, Dr. Tangkuampien suggests that a three-pane view is the most useful. The three-pane view describes the screen and document layout used that presents the viewer with the question, student answer and model answer in a single view. During his research Dr. Tangkuampien has found that this view has proven the most beneficial to both the marker and the student. It allows the students to easily assess what they did incorrectly and how they can fix it. The software shown during the interview maintained the traditional view of a test script i.e. as a set of multiple pages exactly as they were scanned with the model answer added to the right column

Dr. Suleman observed that when working with electronic test scripts one does not have to maintain this view. He suggests that it is possible and may be beneficial to tailor the view of the test script to the device on which it will be marked. When changing the view however, the ability to add feedback should still be maintained as Fowles [11] suggests that this is both reassuring to the marker but is also essential if the marking is challenged in the future.

## Summary

Summary of background work

# The proposed solution

ScriptView is an electronic test management system which enhances the current test management procedures by leveraging the advantages of technology at specific stages in the process. These advantages include the rapid processing of large amounts of data, automated error checking and the ability to efficiently maintain a digital paper trail. It is a complete test management solution which aims to reduce the time taken to mark and redistribute tests as well as capture and analyse the results.

The system allows users to scan the paper versions of the tests and mark them on either a web or tablet interface. The scanned tests are automatically sorted and stored on a central server according to the course and test name. This server will be access controlled in order to ensure the security and integrity of the tests.

Once the tests are available on the server, they can be marked on either a web or tablet interface. Each of these interfaces are optimized to offer the most intuitive marking experience given their limitations such as screen size and input type. Both interfaces aim to decrease the time that is required to mark a test while maintaining high level of accuracy.

ScriptView allows for the distribution of marked scripts to the students and for a summary of the marks of a particular test to be sent to a teaching assistant (TA). The format of the email sent to students allows them to quickly asses in which areas they lost marks and thus where they need to improve. The summary sent to the TA is in CSV (Comma-Separated Values) format which is compatible with most learning management systems and thus reduces the time needed to capture these marks on another system.

Diagram for system and what I’m doing

# Design and Implementation

## Software Development Methodology

A software design methodology is …

## User Centric Development

## Process Design

ScriptView is intended to enhance the existing test management procedures as described in section 2.1. This means that the system needed to be designed in such a way that it would not disrupt the current operations to a large degree.

Figure 2 shows the overall process flow of ScriptView. This flow was inspired by the existing test management systems but decreases the amount of human intervention required.



Figure 2: Overall process flow of the ScriptView system

The first step in the process is preprocessing. The preprocessing step ensures that all constraints on the physical test paper (see section X) are adhered to before the scripts are scanned. These checks are performed by the user but it is not time consuming since the appropriate templates for the test papers are provided and if these are used, this step can be skipped by the user.

Once scripts have been collected and scanned, they are automatically emailed to a monitored inbox. As scripts arrive at the inbox, they will undergo Optical Character Recognition (OCR) in order to determine the name of the course and the test so that they can be stored on the server. This was done to remove the need for users to manually enter this information for each test which decreases the amount of work by the user as well as reducing the probability of naming errors since all text output from the OCR methods is checked a database of test and course information. In addition to the OCR, the digital copies of the tests are also undergo image processing in order to improve their readability when they are displayed on the tablet interface. Once both the OCR and image processing has been completed the script is uploaded as a set of images, each representing a single page from the script.

Ethical concern: Image processing on tests

The act of marking a test script is largely unchanged in terms of where it fits within the process flow and how it is performed. This was done intentionally in order to decrease the amount of changes that would have to be made to the existing test management procedures if an institution wishes to adopt the use of ScriptView.

Test marks are automatically gathered, stored and analyzed by ScriptView. These marks can then be accessed by administrative users and sent to students along with their test scripts. Marks can also be exported in Comma-Separated Values (CSV) format for uploading to the institutions learning management system. (How is this different from the ScriptView section?)

## Initial Software Design

Since an agile methodology with a focus on user feedback was followed during the development of ScriptView. The final product is, in some respects, vastly different from the original design. Therefore this section only documents the initial design of the software before any implementation or user feedback sessions was undertaken.

The components of ScriptView discussed in this paper can be clearly divided into three sections namely test script processing, memorandum processing and the mobile marking application.

### Test Script Processing

Test script processing involves all tasks that are required scan the scripts; retrieve information from the scripts for storage; converting them to a suitable format for the marking interfaces and converting them into a suitable format for emailing to the students.

Two approaches were considered for scanning the test scripts. The first was the creation of a dedicated desktop application which would allow the user to select a set of tests from the local storage on the machine and upload these to the server. This would allow the operation to be strictly access controlled and thus easily traceable to a single person. The disadvantages of this approach however is that the scripts would first need to be scanned and saved to the local machine, which is not only time consuming but creates a single point of failure. The uploading would also be delayed if the user forgets to start the processing on the local machine and becomes preoccupied with other administrative tasks.

The second approach was to have the scripts automatically emailed to a monitored inbox as they were scanned. This allows for the work to be distributed among multiple individuals by allowing different people to do the scanning while still keeping the processing centralized. The process would still be traceable as each documents has a timestamp that can be checked against who was using the scanner at that particular time. The machine doing the processing, is still a single point of failure but the scanned scripts will not be lost as they will still be available in the email inbox and will be processed as soon as the applications resumes.

## Constraints

ScriptView is intended to enhance an existing procedure and as a result certain constraints needed to be established. This was to ensure that the system would be easily adopted by potential users as well as decrease the amount of changes that would have to be made to their current workflow. This section describes theses constraints and how they arose.

### Scanned Test Script

During the design of ScriptView, sample test scripts were provided by the project supervisor. These scripts were previous tests that were used in a single course. Figure 3 shows the variation that exists in the formatting of the first page of each of the scripts.



Figure 3: Variation on cover pages

This variation meant that the scanning component of ScriptView would need to be able to handle all of them as well as any unseen cases. To accomplish this OCR was performed on each cover page and the resulting text was searched for keywords. These keywords were related to the information which needed to be extract for example name, surname or test name. This approach had limited success. The location of fields that needed to be completed by the test taker such as name and surname were accurately detected but the handwriting recognition which was then performed on these regions was unreliable and inaccurate. Text that was typed by the educator such as test name was accurately detected by the OCR but determining whether or not text was part of the test name required that some intelligence be incorporated into the algorithm and this was beyond the scope of the system.

In order to overcome these challenges it was decided that the formatting of the first page of the test would need to be constrained. This would allow the OCR algorithms to categories recognized text based on their position on the page and relative to each other. An example of this is that the algorithm could now determine that text after the institution name is the name of the course.

Keeping the goal of reducing changes to the current procedures in mind one of the sample tests was chosen as the accepted format and it only constrains the top half of the first page. The chosen format can be seen in Figure 3 highlighted in green.

### Memorandum

The memorandum for any test that will be marked using ScriptView needs to be uploaded via a web interface in Portable Document Format (PDF). This enables the system to manipulate the way in which the questions and answers are displayed on the marking interface. In order to accomplish this, the text of both the questions and answers needed to be extracted as well as the location of the answer regions.

Extracting the text was accomplished using a similar technique as was used on the first page of the test. However, instead of using OCR the text is simply extracted from the document. This is possible since the memorandums must always be typed and are not scanned documents. Since the length of both questions and answers could vary and neither the question nor answer positions were known categorizing the text as on the first was not effective.

To mitigate this problem, the use of annotations on the memorandum was explored. This approach involves having the educator place annotations at certain location on the test paper which would indicate the start and end of an answer region. The annotation that was used to indicate the start of an answer was the mark indication which was already a part of the test. To indicate the end of answer a new annotation would need to be created. An example of these annotations can be seen in Figure 4.



Figure 4: A question and answer with the appropriate annotations

These annotations allowed for algorithms which could accurately categorize text into either being a question or answer based on their location relative to the annotations. These annotations were also used to extract the start and end y-coordinates of the answer regions which were needed for mark allocation on the marking interface.

Although this technique was effective, through testing, it became apparent that the spacing on the memorandum and the test script that is handed to students is not always identical. This meant that the coordinates for the answer regions which were extracted would not always be accurate and would as a result skew any analytics performed on the test results.

To avoid this inaccuracy it was decided that the user would need to upload both the memorandum and the test script as it is handed to the students, both in PDF format. Using these it was possible to categorize questions and answers without the use of the annotations as answers were not present on the test handed to students so categorizing text could be achieved by comparing the extracted text from both documents. Answer region detection is now performed on the test as it is handed to students by using image processing to detect the lines left open for a student to write the answer.

The use of annotations was thus discarded but it was now necessary for the user to upload two documents instead of only the memorandum and the space left for students to answer needed to be indicated by lines as seen in Figure 5. Figure 5 also indicates the accepted location for the mark indication.



Figure 5: The accepted format for a question

### Target Device

The final version of the mobile marking application needed to run efficiently on a Samsung Galaxy Note 10.1 and use the Samsung S Pen stylus as input. The operating system on the device was Android Version X (Jelly Bean).

The key constraints introduced by this device was making effective use of the 10.1” (255.8mm) screen and managing the memory when handling a large amount of images.

### Summary

Is this necessary?

## Project Planning

With four months to fully implement the system and the goal of involving users in the development process, a clear plan needed to be established for all phases of the project. Since the agile methodology was going to be used, the project was divided into four iterations. At the end of each of the first three iterations, there would be an opportunity for potential users to provide feedback on the current state of the system. This feedback could be on any aspect of the system irrespective of whether or not they had been fully implemented.

Tasks within each iteration were prioritised based on their importance in terms of achieving the overall goals of the software such as improved marking efficiency and reducing the probability for error. User feedback also influenced the priority of tasks as highly requested, yet in scope, features were prioritised.

## Project Feasibility

Scanning at 300dpi

## Core Functionality Development

# References

* 1. Poor, D. D. (1995). Image Capture and Storage Techniques in Association with Optical Mark Reading
  2. Christie, J. R. (1999). Automated essay marking-for both style and content. *Proceedings of the Third Annual Computer Assisted Assessment Conference, Loughborough University, Loughborough, UK*
  3. Pulman, S. G., & Sukkarieh, J. Z. (2005). Automatic short answer marking. *Proceedings of the Second Workshop on Building Educational Applications using NLP,* 9-16.
  4. Thomas, P. (2003). The evaluation of electronic marking of examinations. *ACM SIGCSE Bulletin,* *35*(3), 50-54.
  5. Siemens, G., & Gasevic, D. (2012). Guest editorial-learning and knowledge analytics. *Educational Technology & Society,* *15*(3), 1-2.
  6. Wiliam, D., & Black, P. (1996). Meanings and consequences: A basis for distinguishing formative and summative functions of assessment? *British Educational Research Journal,* *22*(5), 537-548.
  7. Forehand, M. (2010). Bloom’s taxonomy. Emerging Perspectives on Learning, Teaching, and Technology, , 41-47.
  8. Aboulsoud, S. H. (2011). Formative versus summative assessment. *Education for Health (Abingdon, England),* *24*(2), 651. doi:651 [pii]
  9. Alisi, T. M., Del Bimbo, A., & Valli, A. (2005). Natural interfaces to enhance visitors' experiences. *Multimedia, IEEE,* *12*(3), 80-85.
  10. Malizia, A., & Bellucci, A. (2012). The artificiality of natural user interfaces. *Communications of the ACM, 55*(3), 36-38.
  11. Fowles, D. (2005). Literature review on effects on assessment of e-marking.