

An Exam Scoring Application Running on Paper Architecture

Narcis Lozano, Masaki Nakagawa †

†Tokyo University of Agriculture and Technology, Naka-cho 2-24-16

Koganei, Tokyo 184-8588, Japan

E-mail: narcis@hands.ei.tuat.ac.jp

Abstract This paper presents a scoring system for exams fulfilled using pen devices that can write on paper. First, the students' handwriting is stored as online data, and then processed in a server using a software architecture (Paper architecture) that allows the use of several different types of pen devices without having to reprogram the main application. Questions are scored using a hybrid system, with automatic scoring for test questions and a human assisted scoring environment for free-text answers. Finally, test scores are calculated and the results sent to students by e-mail.

Key words InkML, lazy recognition, pen devices, character recognition, e-learning

1. Introduction

In recent years the number of models of PDAs, tablet based PCs and tablets has increased. In addition to this, new types of pen devices have been developed. Among these, the new devices that can be used to write on paper instead of a tablet or a computer screen have been drawing increasing attention.

This kind of devices usually consists of a pen and a device that can track the position of the pen and store its path in memory [1] [6]. There are also other pen devices that can calculate its position by using a special paper printed with dots [2]. This data is sent when the device is connected to the computer and then processed.

Pen devices that use paper replace the computer keyboard and mouse, making the process of fulfilling documents more natural to the user, especially for people not accustomed to use computers. The fact that at the same time that the data is inputted in the computer a handwritten copy of the paper is also generated makes these devices especially suitable for some purposes.

On the other hand, because the data is not processed until the device is connected, the interactivity of applications for these devices is comparatively low. This is

especially noticeable for the correction of recognition errors. With tablets errors can be detected as the user writes and corrected interactively. In devices that use paper, however, errors usually cannot be detected or corrected until the device is connected to the computer.

In this paper we present an exam scoring application using pen devices that can write on paper. Although there exists test scoring systems that use scanned images [3], these are often limited to Boolean or answer picking questions, and because of the difficulty of segmenting foreground and background in images, they usually require the answers to be fulfilled in a separated sheet of paper, a circumstance that can be confusing for the student.

Pen devices segment automatically the track of the pen from the background, require far less storage capacity than images and can retrieve additional information such as stroke order or pressure. Our scoring application supports both automatic scoring for test type questions, and human assisted scoring for free-text questions. Additional features have been developed to reduce the scoring time.

The system runs on a software architecture that supports several types of these pen devices without having

to reprogram the main application. Test results are sent directly from the correction application to the students by e-mail.

The structure of this paper is as follows. In section 2 we present Paper Architecture, an unified software architecture for pen devices. In section 3, we describe an exam scoring application developed using Paper Architecture. Section 4 details results and future work and in Section 5 we draw the conclusions.

2 . Paper architecture

The system presented in this paper runs on a unified software architecture that we will refer as Paper Architecture. The main goal of this architecture is to allow using the same application with different pen devices. As was explained in the introduction, there are several types of pen devices that can be used with paper, each employing different technologies, software interfaces, coordinate systems, etc. No standard has been defined for the communication between these devices and the computer and none of the current devices is predominant enough to be considered a de facto standard. This means that developers have to reprogram their applications for each kind of pen device they want to support.

We overcome this difficulty by creating a software architecture that abstracts the communication between pen device and computer using the concept of lazy recognition [5], that is, the processing and recognition is performed after all the handwritten data has been inputted.

2 .1 Data capture

A client application is created for each pen device that we want to use in our system. This client gets the on-line data from the device in the proprietary format. Then it segments the data into pages, when this is applicable to the device. It then proceeds to map the coordinates to a standard coordinate system. Finally it stores the data in an InkML based file. The process flow is shown in Figure 1. These operations require only the most basic functions of the pen device, and therefore these clients can be developed easily .The files are then sent to the server, where they become the input data for the main

applications.

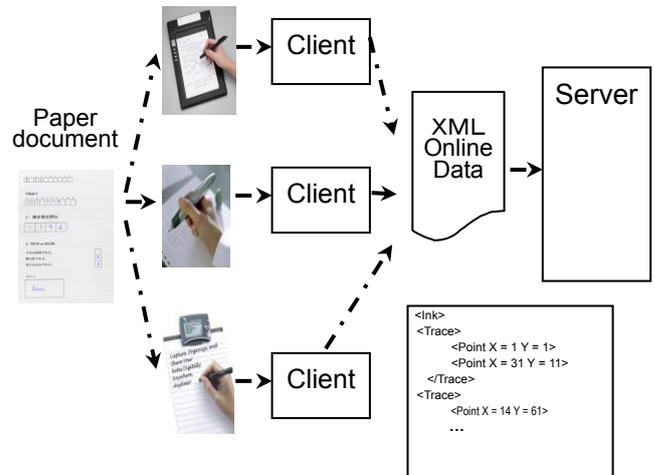


Figure 1 Data capture

2 .2 Data processing and output

The processing of the data takes place in the server, the flow being shown in Figure 2. The data file that receives the server contains only streams of point coordinates. In order to process the data, the definition of the structure of the document (position and type of the data, etc.) is needed. To this purpose, template files are used. These are stored in the server in order to improve the security and the maintenance of the applications. To decide which template must be used, an especial area in the document is reserved for the user to write an identifier. When the data arrives to the server this area is segmented, the content recognized and the identifier used to retrieve the corresponding template.

Templates are XML files containing a definition of the structure of the document and associated data. In order to improve reusability and simplify the design, the structure is defined using objects with associated functionality: questions, text input fields, drawing input fields, etc. The template is created interactively using a CAD tool to add the necessary objects.

Using the template, the data is first segmented, associated with the suitable object and then processed according to the parameters of the object. The results generated by the different elements are then grouped and exported as an XML output file. This output file can be used to update a database in the server or sent to the user to be employed as an input to another application, such a visualization tool, a web browser, etc.

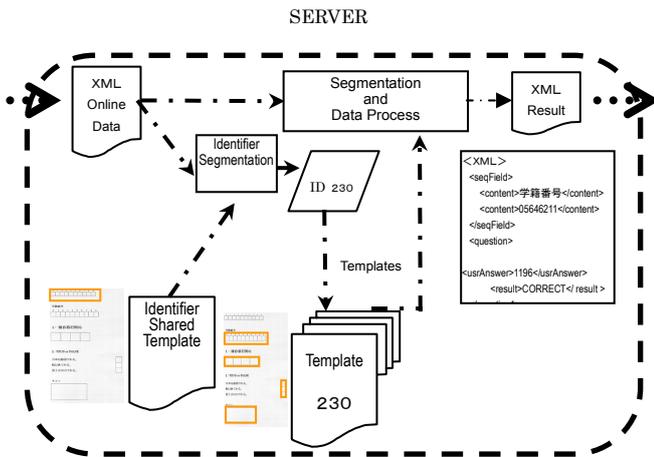


Figure 2 Data process

3. Human assisted exam scoring application

In order to test the capabilities of Paper Architecture, a human assisted exam scoring application was developed. This application is aimed to assist teachers to correct and score exams, and to analyze and store the student results.

Most research in exam scoring interfaces has focused in fully automatic scoring. The user input must be segmented, converted to text, and then compared to the answer to decide if it is right or not. This approach has several problems. Cursive handwritten character recognition is an error prone process, and errors in segmentation and recognition may lead to falsely evaluate a question as incorrect. In the case of interactive systems that use tablets the user can detect these errors and rewrite the pattern. However pen devices that write on paper lack of feedback, and therefore it should be the teacher who would have to correct all errors before evaluating the questions. When the number of students and questions is high, such a system becomes impractical.

On the other hand questions normally can have more than one possible answer. In automatic scoring systems, the teacher should guess beforehand all possible answers, which is often impossible. Also, in the case of free-text answers, the system should be able to fully understand it, in order to evaluate it. Despite it has been research in automatic correction of this type of questions [4][7], these methods are really meant for evaluation result estimation, not for unassisted correction.

Because of these problems, most exam scoring systems

deal only with test type, Boolean questions. However, with this kind of questions, exams become a selection process. Users only have to be able to decide if an answer is right or wrong, but may not learn the contents thoroughly. For example, in the case of language tests, the user may learn to discern the meaning of vocabulary, but not to spell it correctly. On the other hand, it becomes difficult to test the reasoning and expression abilities of the student, being this type of question appropriate mostly just for memorization based contents.

In this scoring system we have taken a hybrid approach. Automatic scoring has been implemented for test type questions and for question that the teacher considers that all possible answers can be limited (dates, country capitals, for example). For questions that can have multiple answers, or that require to write free-text answers, a human assisted scoring interface is used. Instead of recognizing the handwritten data and converting it to text, we store the online data as streams of coordinates. These are later shown as images, so the teacher can read them and evaluate the answers, using an interface designed specifically for this purpose. This process is explained in detail later.

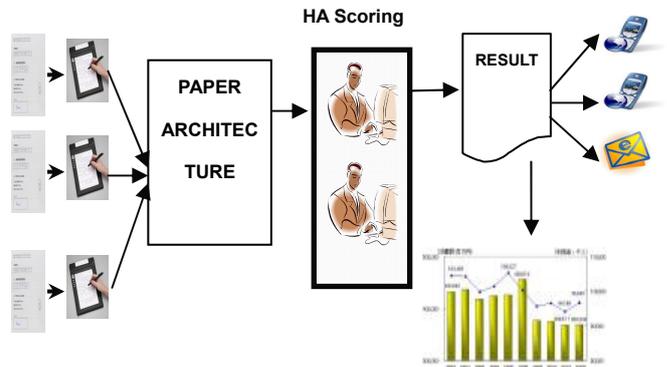


Figure 3 Exam scoring process flow

The exam creation process is as follows. The teachers first write the questions of the exams using a text processor, and print it. Using this printed document as a guide, the teacher creates the template file adding question elements with the CAD tool, and assigning the value of each question. For answers that are scored automatically the correct answer is assigned to the question as a text string. For questions that will be evaluated using the human assisted scoring interface the

correct answer may be given as text as well as inputted directly as handwritten data. Finally, the template or templates are stored in the server, and an identifier is automatically assigned to them. The teacher then proceeds to create copies of the exam paper, which are then distributed to the students.

The flow from the fulfilling of the exam to the delivery of the result is shown in Figure 3. The students fulfill the exam using the pen device. They first write the exam identifier, which is given to them by the teacher, in the especial area reserved for this purpose. They also fill in another especial area with their own student identifier, which in our example was the student registration number. The students then proceed to fulfill the answers. When they finish the exam, the data is downloaded one by one using the client application, converting it to a common coordinate system, and storing it in an XML file. These XML files can be sent one by one or all in one package to the server. The printed copies of the document fulfilled by the students is also gathered and stored, to compare them with the computer scored exams in case of errors.

In the server, the identifier of the exam is segmented and recognized, and the corresponding template is retrieved. Using the information about the document structure of the template, the server segments the handwritten data of the answer to each question. In the case of the questions that are to be scored automatically the answer is converted to text or to a Boolean value, compared with the correct answer and evaluated. The rest of the questions are stored as handwritten data in InkML format. The answers and the student identifier for each exam are stored in an XML file and sent to the teacher for the human assisted scoring stage.

In the human assisted exam, scoring stage the teacher uses an application to correct the answers that were not scored automatically. First, the teachers load the exam template file in the application, which contains the value and correct answer for each questions. They then proceed to load the files of the student answers that were processed by the server. They can then verify if the answers that were scored automatically have been done correctly so.

For answers that have to be scored manually by the



Figure 4 Human assisted scoring interface

teacher, an interface was developed for this purpose. The objective of this interface was to allow the teacher to correct and evaluate the answers as fast and comfortably as possible, trying to retain all the possibilities that offered scoring using paper and pen.

The interface used is shown in Figure 4. On the left of the screen a tree view of the student exams allows the teacher to access hierarchically the answers ordered by question number or student. Once the question has been selected, the answers are shown in the screen. In the bottom part of the screen, the correct answer is shown either as a text string or handwritten data, depending of the way it was inputted in the template design step. On the right of each answer there is a small interface for evaluating the answer: one button evaluates the answer as correct, giving the maximum score, another as incorrect, and with the others as partially correct, assigning it only a fraction of the maximum score. There is also a text box in case the teacher wants to give the answer a more precise scoring. The teachers can also write over the answer using the mouse or a pen device, as they would do on paper. This correction ink information is displayed in red and is stored as stream of coordinates along with the answer shown in Fig. 5..



Figure 5 Adding correction notes with pen device

The time needed to correct the exams can be reduced taking advantage of the fact that the answers are not bound to the paper, and can therefore be reorganized and sorted according to the teacher needs. When scoring paper exams, teachers must correct one after another. They have to look which question is, think the correct answer, look the student answer and evaluate it. Our system allows using this way of scoring, and in addition it provides another, more effective when there are many exams to score (Figure 6). Instead of scoring all the questions for one student, we group all the students' answers by question. The teacher first corrects all the students' answers for question 1, then all questions answers for question 2, etc. Organizing questions in this fashion, which is usually no possible for paper exams, teachers do not have to be constantly bothered thinking which question they are scoring or which is its correct answer, and can focus their attention in the student answer instead.

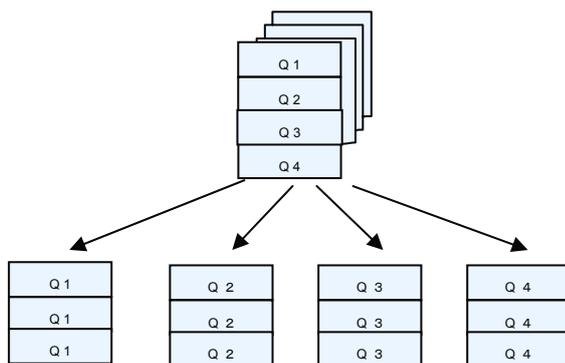


Figure 6 Question reorganization

Additional features have been implemented to further reduce the scoring time. The teacher can decide to automatically evaluate the questions that have not been answered by the student as incorrect. In many exams, the amount of questions not answered can represent an important percentage of the total, so the reduction of scoring time can be considerable. He also can hide the answers that have been already scored so that they can focus their attention on unanswered questions.

Once all the answers have been evaluated the teachers can export the results. These are presented in a spreadsheet and contain global statistics of the exam as well as of each student. This way the teachers can generate charts easily to visualize data about the progress of each student, and the difficulty of each question. Additionally, importing contact data of the students, the test results can be sent as e-mail to the students' cellular phone or e-mail account, shortening the time between the student finish the exam and receives the evaluation.

Finally, in contrast to other type of evaluation systems that use computers, if for some problem the data that fulfilled the student is lost, a copy in paper of the exam in paper is still available. This may also be useful for eventual reclamations of students for misrecognized answers.

4 . Results and future work

The prototype system was implemented and tested in small scale exam simulations. In these test Digimemo pen [1] devices were used. A short exam, consisting mainly of questions about English vocabulary questions and fill-in-the gaps questions were fulfilled, processed by the server, and then processed interactively with the assisted scoring interface. Finally, exam results were exported as spreadsheets for analysis and sent to the student e-mail address.

The main problems have been answers that were not fully captured by the device, or that were not correctly segmented because the paper was moved by the user, and therefore the areas in the document did not correspond to the ones in the template. Since this system is intended for human assisted scoring, and for most questions it does not perform character recognition, the first type of error usually still makes possible the correct evaluation of the

question. However, the second type of error mixes answers together and often makes impossible the scoring. One advantage of this system is that the original exam paper is available for comparing and evaluating manually when errors arise.

The test identifier and the student identifier are inputted ticking in order in a set of drawn squares as shown in Fig. 7, which have a number associated, in a way similar to a keyboard. This is possible because pen devices, unlike image based systems preserve time information for the strokes, so the order of the ticks can be known. This method is more reliable than recognizing handwritten digits.

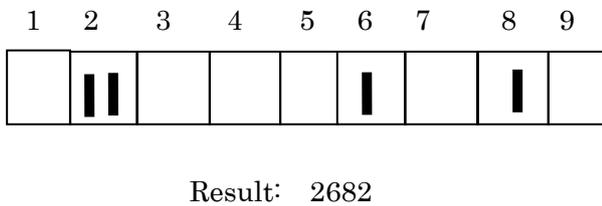


Figure 7 Identifier input

The results generated by the exam allowed creating exhaustive statistics and graphs. The system stores not only the global score of the exam, but the individual score for each question. This is an improvement over the traditional way of scoring exams, where usually the later information is discarded because of the bulk of having to input all this values manually in the computer. It was also possible to quickly reevaluate the exam when a question evaluation was changed (which would be necessary if a student makes a reclamation) or the score of a single question was changed (for cases that a question must be annulled for all the students), which if we used the traditional method of evaluation would require extensive recalculations.

In the future, we plan to test the system for a greater number of students and questions, in order to fully evaluate the potential of this system to shorten scoring time and showing statistics in more meaningful ways. We also plan to create a more robust method of segmentation, using the exam template for analyzing the document structure, in order to reduce segmentation problems due displacements of the paper.

5. Conclusion

In this paper we have presented a hybrid exam scoring application using pen devices that can write on paper, which uses automatic scoring for test type questions and human assisted scoring for text-free questions. Question reorganization, one-click question evaluation, detailed statistics export and automatic e-mailing of test results to the students were also implemented

The system runs on a software architecture to allow to be used with different types of pen devices without modifying the original application. Future work includes evaluation for a larger number of students, and improvements in the segmentation process.

6 . References

- [1] Acecad website. Available at <http://www.acecad.com.tw>
- [2] Anoto website. Available at <http://www.anoto.com>
- [3] Gorgevik D. , Mihajlov D. Grcevski N. , "A Simple System for Automatic Exam Scoring Using Optical Mark Reader", *Proceedings of Second Balkan IFAC International Symposium APPLIED AUTOMATIC SYSTEMS AAS'2000*, AAS7-3, 21·3 September, Ohrid, Republic of Macedonia, 2000.
- [4] Burstein, Jill. et al. "Automated scoring using a hybrid feature identification technique" International Conference On Computational Linguistics. Proceedings of the 17th international conference on Computational linguistics - Volume 1 Pages: 206 – 210. Montreal, Quebec, Canada 1998.
- [5] Nakagawa M., Machii K., Kato N., Souya T. "Lazy recognition as a principle of pen interfaces". INTERCHI'93 Adjunct Proc. 89-90. 1993
- [6] Pegasus Notetaker website. Available at <http://www.pc-notetaker.com>
- [7] Wild, F., Stahl, C., Stermsek, G., & Neumann, G. (2005) "Parameters Driving Effectiveness of Automated Essay Scoring with LSA". In *Proceedings of the 9th International Human assisted Assessment Conference (CAA)*, 485-494. Loughborough, UK, July, 2005.