# Premark: A System Designed To Organising Course Work For Assessment

Ambikesh Jayal*, Michelle Cartwright, Martin Shepperd

## ABSTRACT

*Unfortunately a side effect of the introduction of some e-learning systems into HEIs has been the loss of flexibility and the imposition of new processes that can result in additional clerical and administrative burdens. In this paper we describe the development of a software tool named PreMark that fits between the e-learning tool (in our case WebCT) and the user (lecturer). PreMark is a prototype software system designed to provide support for lecturers during the assessment process by organizing the coursework electronically submitted by the students. Additionally it helps detect possible cases of plagiarism. Much time can be spent arranging the coursework submitted by the students before actual assessment can take place particularly when there are large numbers of students enrolled on the course. The problems increase when some students do not exactly follow submission instructions. We argue that there can be significant benefits in terms of reduction of effort and in plagiarism detection from such software which can be developed quickly and cheaply as middleware components that interface between lecturers and e-learning systems.*

***Keywords:*** Assessment (e-assessment), marking, elearning (e-learning), coursework.

## 1. Introduction

Lecturers can spend a considerable amount of time arranging the coursework submitted by the students via an e-learning system such as WebCT before they can start marking. The lecturer needs to go to each student's coursework, download it, open it, and print it, which takes a considerable amount of time and can be quite troublesome especially if the there are many students in the class. The problem increases when some of the students have not followed the guideline to write their student identification number in the coursework, in which case the lecturer needs to insert it in the coursework before taking the printouts. Also lecturers spend lot of time sitting before computer screen to mark the coursework. So many lecturers might prefer a paper version of the coursework. A solution is to create a middleware component that creates a consolidated output file of all the students' answers as a single PDF or word document, so that the lecturer can take the printout in one go and mark them at their convenience without the need to be in front of computer screen while marking. Our PreMark tool basically does that. Additionally PreMark has some additional features like organise by students' id and organise by questions and organise by similarity.[1]

## 2. Related Work

Stephens (Stephens, Sargent et al. 2001) has summarized the requirements of an ideal Integrated Marking

[1] School of IS, Computing & Maths, Brunel University, United Kingdom

* *Corresponding Author:* (Email: ambikesh.jayal@brunel.ac.uk, Phone: +44-7882421958)

Tool which includes features like intuitive marking devices, detecting plagiarism, automatic grading, report generation etc. We found three existing Marking systems MarkTool (Zhang, Heinrich 2005), GradeMark (GradeMark) and Classmate (Baillie-de Byl 2004), All the three systems are onscreen marking tools with some valuable features like the PDF annotation, edit and mark online, integration with some electronic learning environments like webct (WebCT). But none of the systems parses all the individual coursework and produces one consolidated PDF and so the many clicks to reach student's coursework and download, open and print it are still unavoidable. Also the "organize by questions" (all answers to a particular question arranged sequentially one after another in a single file) and "organize by similarity" feature missing in all of them.

Our tool is different in the way that it organises the coursework of students so as to make it as close as possible to the old way of paper and pen marking style by parsing all the individual coursework and produces one consolidated PDF document. Apart from that there are some additional features in our system like "organize by questions" (all answers to a particular question arranged sequentially one after another in a single file) and "organize by similarity", which might help detect plagiarism cases as well.

## 3. PreMark System

The PreMark system is a middleware component that creates a consolidated output file of all the students' answers as a single PDF or word document, so that the lecturer can take the printout in one go and mark them at their convenience without the need to be in front of computer screen while marking. To capture the requirements for the PreMark system, we have used an iterative methodology which included few iterative cycles of meetings and building prototype. The following is the list of requirements captured for the PreMark System:

- Integration with e-learning management systems:
  The middleware component should be integrable with the e-learning management system which in our case is Webct. Webct allows downloading of all the students submitted course work as a single compressed zip file which contains folders for each student with name as their student identification number, within each of which the respective students coursework is stored as a single compressed zip file. The middleware component should take a compressed zip file as the input and produce a single PDF document containing all the students' answers in a well presented manner, so that the lecturer can take the print out of that PDF document and mark away from a computer if so desired.
- Customizable:
  The lecturer should be able to choose the type of output file required which could be PDF, Ms Word Document, Open Office Document etc and the header and footer content of each page of the output file which could be the students' identification number, students' name, file name, time of submission, question number etc. Also the lecturer should be able to select the type of the image file compression required in the output file. JPEG is one of the widely used and efficient image compression formats and so by default the component should convert all the images (like wmf, gif, tif etc.) into jpeg images before inserting them in the output document.
- Resizing Images in Coursework:
  In the coursework involving the images, the students sometimes submit images of large width and height size which might not fit well in a single screen on the machine used by the lecturer and so the lecturer needs to resize these images for viewing, marking and taking printouts.
  So for the coursework involving images the middleware component should be able to automatically resize images so that they can fit on a single screen and can be printed on a single paper.
- Question Organisation:
  The coursework in the output file should be organised as per the following parameters:

- Organise by student identification numbers: In this case the coursework should be organised in the output file in the order of the student identification numbers.
- Organise by question number: In this case the coursework should be organised in the output file in the clusters of answers as per the question number i.e. all the answers for first question should be at one place, followed by all the answers for the second question and so on. Moreover an additional document should be generated containing all the student identification numbers with the questions numbers against them, so that lecturer can enter marks for questions for students. The reason behind this requirement is that sometimes it may be easier for the lecturers to mark all the answers for a particular question at a time one after the other. This feature may also be help when dividing the coursework for marking among lecturers based on the question number so for example all answers to question one are marked by a lecturer and all the answers to second question are marked by the another lecturer. If the student submits all the answers in a single document then one of the obvious problems while parsing the student answer document would be to find beginning and ending of each answer. One of the possible solutions for this can be to issues answer templates to the students with markers for various questions.
- Organise by similarity: In this case the coursework should be organised in the output file as per the similarity index. This can help detect plagiarism as well. One of the naïve criteria for clustering the students' answers can be on the basis of the size of the student answer files. So say all the students' answers with the file size of 500-600 Kb would be in one cluster while those with size 600-700 would be in another cluster and so on. In the prototype developed we have applied this criterion for clustering. We are currently working on the other approaches to determine the similarity index.
- Report Generation: A report should be generated for the processing done on each coursework. Since some coursework files could be corrupt or the system might not be able to open them properly so there is likelihood or some coursework left unprocessed. In this case a separate error report should be generated containing the information about the unprocessed coursework. Also a Similarity report for clusters should be generated like in case of cluster by file size; the report should contain all the clusters that have more than one coursework.
- Email Output documents: The component should be able to save the work on a central server and should automatically email the output documents to a group list email or individual emails.

## 4. Implementation

We have developed an initial prototype of the PreMark system with limited functionality. The PreMark system is platform independent and all the technologies used for the development of the PreMark prototype system are Open Source Technologies (Open source) namely the programming language (Java) and the free integrated development environment NetBeans IDE (Netbeans). Apart from this the iText API (iText) has also been used which is an Open Source library for dynamically generating the PDF documents and includes options to set the header, footer and font of the PDF document. The prototype has been implemented for an undergraduate module with about 240 students at Brunel and took 13 minutes and 9 seconds to run and successfully produce the consolidated output file in PDF format.

## 5. Discussion and Conclusion

The paper has considered the problem of time spent by the lecturers in organising the coursework and presented a cheap and easy middleware component to help lecturers' organiser the coursework more quickly. The time thus saved can be spent on the actual marking process and can be quite helpful particularly in those cases where there are many students in the class and there is a time constraint on lecturers to mark and provide feedback to the students.

The effort in terms of time in developing the PreMark prototype system has been about 150 person hours. Apart from this there has been little cost for the software and the development environment because all the software used for the development has been Open Source technologies except for the Operating System Windows Professional XP which in this case was provided for free by the developing university. The initial remarks of the lecturer regarding the test run of PreMark were encouraging. We are in the process of doing a qualitative interview with the lecturer to gather the experiences and effectiveness of running the PreMark and would be reporting it later. So we conclude the following points:

- Cheap and easy middleware can be helpful.
- Simple approaches can be helpful for flagging potential cases of plagiarism.

## 6. Future Work

We intend to develop the PreMark system prototype into a complete system. We also intend to go beyond the boundaries of the developing institution and provide it for use at other institutions for which we need to do the proper documentation of the system. Also we are exploring more effective ways of finding the similarity index of coursework so that coursework can be clustered as per the similarity index, which might be helpful for flagging potential cases of plagiarism. Additionally we intend to carry out experiments evaluating the use of Table PC (TabletPC) by the lecturers for marking the coursework and automatically capturing the feedback by lectures.

## References

1. Baillie-De Byl, P., (2004). An Online Assistant for Remote, Distributed Critiquing of Electronically Submitted Assessment. *Educational Technology & Society*, 7, pp. 29-41.
2. GRADEMARK, turnitin. Available: http://turnitin.com/static/grademark.html [February, 2007].
3. ITEXT, iText, a free java-PDF library: home page. Available: http://www.lowagie.com/iText/ [February, 2007].
4. JAVA, java technology. Available: http://java.sun.com/ [February, 2007].
5. NETBEANS, welcome to NetBeans. Available: http://www.netbeans.org/ [February, 2007].
6. OPENSOURCE, open source initiative OSI - welcome. Available: http://www.opensource.org/ [February, 2007].
7. Stephens, D., Sargent, G. And Brew, I.,(2001) Comparison of Assessed Work Marking Software: Implications for the Ideal Integrated Marking Tool *(IMT).*
8. TABLETPC, what is a tablet PC? Available: http://www.microsoft.com/windowsxp/tabletpc/evaluation/about.mspx [February, 2007].
9. WEBCT, WebCT.com. Available: http://www.webct.com/ [February, 2007].
10. Zhang, J. and Heinrich, E., (2005). A System Designed to Support Formative Assessment of Open-Ended Written Assignments. Advanced Learning Technologies, 2005.ICALT 2005.Fifth IEEE International Conference on, pp. 88-92.

### *About the Authors*

*Ambikesh Jayal* is a final year PhD student at Brunel University UK in the School of Information Systems, Computing & Maths. His PhD is in the area of electronic learning and electronic assessment and his other research interests include service oriented architectures and open source software. He is a member of the Brunel Software Engineering Research Centre (B-SERC).

*Michelle Cartwright* is a Lecturer of Software Technologies at Brunel University in the School of Information Systems, Computing & Maths. Her research interests include empirical software engineering and object-oriented metrics and she is a member of the Brunel Software Engineering Research Centre (B-SERC).

*Martin Shepperd* is Professor of Software Technologies at Brunel University in the School of Information Systems, Computing & Maths. His research interests include empirical software engineering and he is director of the Brunel Software Engineering Research Centre (B-SERC).