Blaise – A Tablet-Based Paperless Submission and Grading System for Writing Courses

Rahul Simha[1], Sean Hanlon, Michael Gaiman, Jared Kiraly, Eisuke Arai
Department of Computer Science
The George Washington University
Washington DC 20052

Abstract

This paper describes Blaise, a software/hardware system designed to support writing programs. Students write papers using the Blaise editor, a simple text editor and submit them electronically to a backend database. Instructors download student papers from the database onto a tablet PC and then use pen-like markup to grade the papers. These papers are then returned electronically to the students. Because modern writing programs emphasize working through drafts and peer-review, Blaise maintains multiple drafts for each assignment, and also supports multiple graders. Furthermore, the backend database allows students to maintain a long-term writing portfolio and enables institutions to track and collect data for assessment of writing programs. Blaise is free, open-source software available from the Blaise website: www.blaise.gwu.edu. Because it is written in Java, Blaise can be used on Windows, Mac and Linux.

Keywords

Tablet PC, electronic paper grading, peer review, electronic markup, writing technology

1 Introduction

Computers and internet technology have made their presence in the modern educational institution [2, 4, 7, 8]. Universities and schools both use backend databases for records, and are increasingly using computers and Internet applications in the classroom itself. In addition, related technologies such as PDA's, projectors and smartboards are helping transform the way courses are conducted. One relatively new and growing technology is the tablet PC, which unlike regular PC's is based on a pen-like interface: users use a pen on a touch-sensitive screen to point-and-click as well as to markup documents. This electronic pen has appealed to users in a wide variety of applications, both for its distinctive use as well as for the simple convenience of doing away with keyboard and mouse. The pen-like interface is also a natural for grading term papers – an instructor who must grade a Word document, for example, prefers to print and mark up with a real pen rather than type laboriously into the same document with a differently-colored font.

[1] Contact author with address: Department of Computer Science, The George Washington University, Washington DC 20052, USA. Email: simha@gwu.edu. The other authors are students, in the order in which they joined the project. Some students were supported by NSF REU stipends and some were supported through internal funds from The George Washington University.

Recognizing the value that tablet PC's can provide, and the fact that no free open-source product was to be found in the marketplace, we decided to develop Blaise, a complete software system for writing courses. Blaise is named after Merlin's eponymous scribe in the legend of King Arthur. In consultation with George Washington University's writing faculty, we added features to support modern writing programs[2].

What exactly does Blaise consist of? Blaise consists of three mutually compatible software packages. The first, the Blaise Editor, is what students use to compose their papers; it is a simplified editor with some standard editing features such as justification and font style (bold, italic). Students also use the Blaise Editor to electronically submit their papers. Because Blaise supports multiple drafts, the Editor offers views of previously marked-up drafts, each of which could have been graded by multiple instructors. The second component, the Blaise backend, is a standard relational database that stores papers submitted by students and graded papers uploaded by instructors. The third and signature component, the Blaise Grading Portal and Grader, is the software that runs on the tablet PC and is set up to incorporate both pen-based markup as well as keyboard-driven input. We describe each of these in detail through an example in the next section.

2 How It Works

2.1 Using the Blaise Editor: a student writes a draft

Figure 1 is a screenshot of the Blaise Editor. This is what a student sees upon launching Blaise on his or her PC. The lower half is the "work area" where the student types in their paper. A toolbar in the middle offers several standard document editing features. The top half is currently blank because this is a first draft. If it were a later draft, the top half would show previous drafts (as in Figure 6).

[2] We are pleased to acknowledge the valuable feedback given to us by Profs. Cayo Gamber and Mark Mullen from GW's University Writing Program.

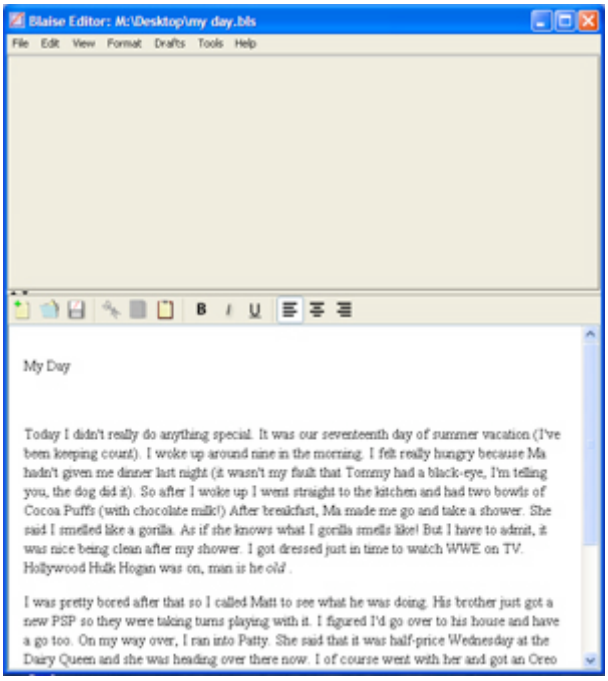

Figure 1: *Student writes a paper with the Blaise Editor*

When the student finishes this first draft, the student finalizes the draft prior to submission. This "finalization" simply indicates that the on-and-off editing process over the past few days is now complete. Finalization is irreversible so that no more changes can be made to that draft at the time of submission. Once the current draft is finalized, the student uses the Editor to electronically submit the paper (see Figure 2). After that, the Blaise backend sends an email receipt to the student, confirming paper submission. Note that graded versions of drafts are stored separately, so the student has the option of viewing the original submission of each draft in addition to viewing the comments made by one of the various graders.

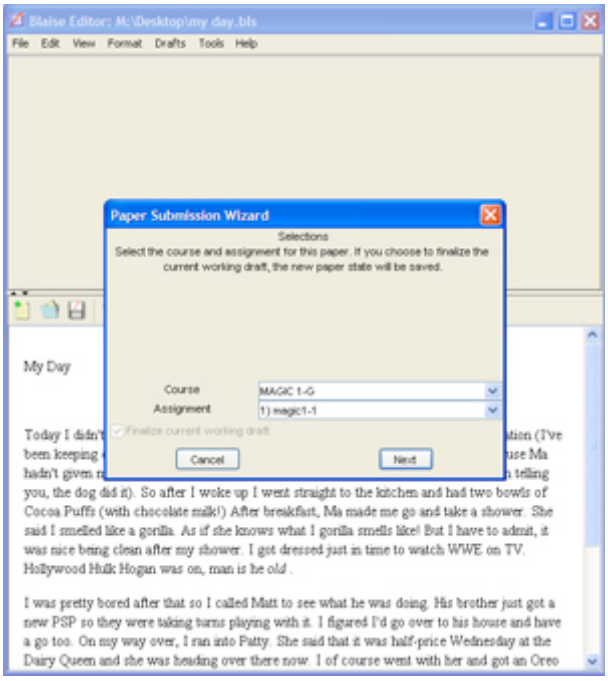

Figure 2: *Student submits paper*

2.2 Using the Blaise Grading Portal and the Grader: the instructor grades a draft

Next, the instructor fires up the Blaise Grading Portal on the net-connected tablet PC in order to check-out and download submitted papers for grading (the login screen is shown in Figure 3). This portal package manages the collections of papers for the instructor. Because it's likely that an instructor will be grading papers in multiple courses, we felt it would be useful to let the software organize the documents by course and to let the software keep track of which papers have been graded and returned to the system. This also has the advantage of freeing the instructor from looking for documents in the file system and keeps the interface portable across operating systems (Windows, Mac, Linux).

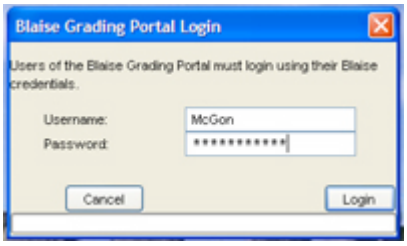

Figure 3: *Instructor logs in to Blaise Grading Portal*

The grader then specifies a course or group of courses to examine what papers need to be graded. The portal thus maintains some kind of workflow, to allow instructors to perform grading in small groups of papers or even individual papers. Once the currently ungraded papers are listed (see Figure 4), the instructor then "checks out" papers to download to

the tablet.

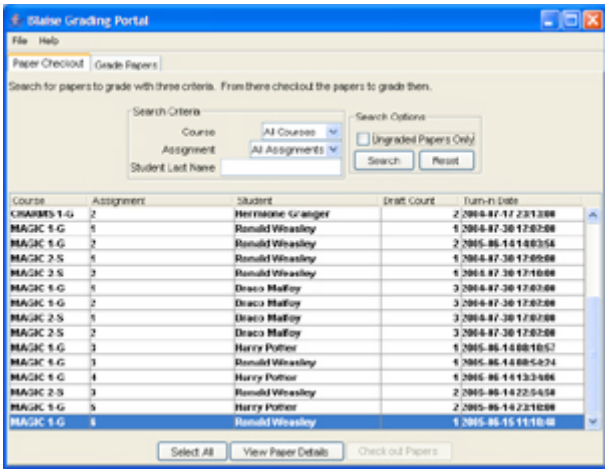

Figure 4: *Instructor downloads selected papers from list of all ungraded papers*

Once ungraded papers are checked out and downloaded, the instructor uses the Blaise Grader to correct the papers (Figure 5). Unlike the Grading Portal, the Blaise Grader does not require an Internet connection, so grading can be done anywhere once download is complete. To mark up the paper, the instructor may insert space, add notes (the yellow post-it notes shown in Figure 5), or write scribbles on the various drafts. Upon completion of grading, the instructor returns to the Grading Portal to check-in the graded papers, sending out email receipts to students, notifying them that their papers have been graded. The instructor may choose to send out email receipts as soon as each paper is graded, or may stage the sending of email receipts so that all are sent out at once.

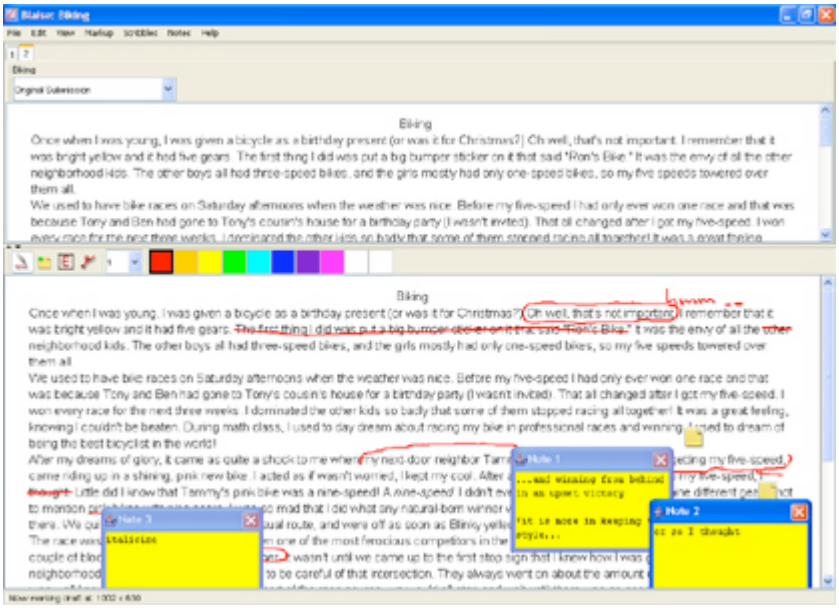

Figure 5: *Instructor marks up paper*

2.3 Students receive their graded papers

Once papers are graded, students receive email receipts notifying them of their newly graded papers. They then download their graded papers from the Blaise website. Figure 6 shows the Blaise Editor after a paper has been through several drafts. The small tabs in the top left corner indicate three completed drafts, of which the second is in view. The horizontal bar below the draft tabs can be used to select one of several graders, if there were multiple graders, and can also be used to view the original ungraded draft. After reviewing the markings, students write another draft and the cycle starts again with a new draft, as shown in the bottom part (below the toolbar).

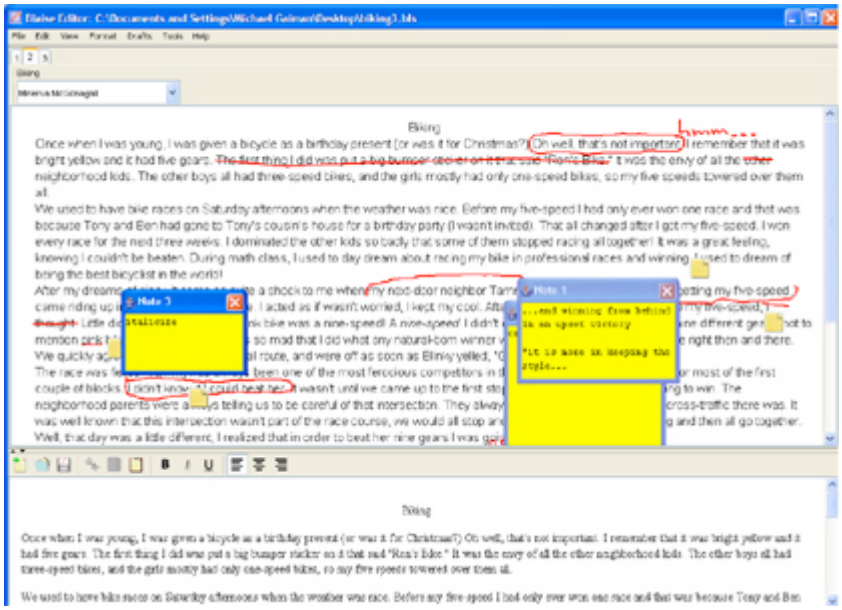

Figure 6: *Student reviews graded paper and starts the next draft*

Implementation Details

In order to be platform-independent, Blaise was written in Java, with the interfaces implemented using Java's Swing library. Thus, Blaise can be run on Windows, Mac or Linux. However, due to minor Java differences between these operating systems, the appearance and look-and-feel might vary slightly. Most of our testing was performed on Windows and Mac (OSX).

Figure 7 shows an architectural overview of the system, with the student working on a PC, the instructor using a tablet PC (lower left) and the backend running on a server that also runs a database server. All the software has been written in Java. Thus, Java servlets are used on the backend for handling communication between student PC's and the

server as well as between the instructor's tablet and the server.

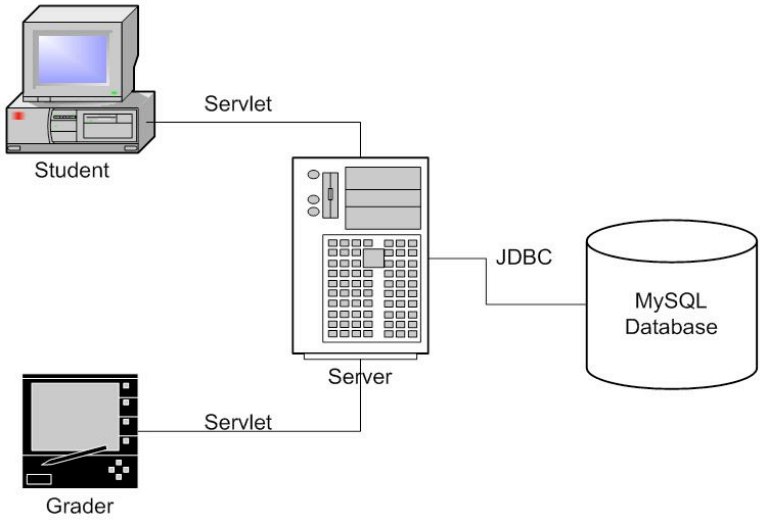

Figure 7: *System view*

The text of a paper written in the Blaise Editor is converted into HTML format for platform independence. When saved, a Blaise paper is put into a directory structure and packaged into a jar file (Figure 8). The jar archival format is used to wrap all the drafts and additional information into a single file for transmission back and forth to the server. This archival format also enables expansion in the future to include other types of grading, attachments and figures. Within the draft subfolder, there are additional subfolders for files and comments (the markup or notes inserted by graders). When a draft is added to a paper, it is copied into that draft's files folder. This copying is done so that changes to the original document will not be reflected in the version put in the paper. The draft has a subfolder for each comment, so that comments from different graders can be viewed separately. Since graders can add space to documents, a copy of the draft is also put into the comments folder. The comment folder also contains two properties files, one for scribbles and one for text notes. Finally, each comment folder also has a subfolder for files, in case the comment is an external document.

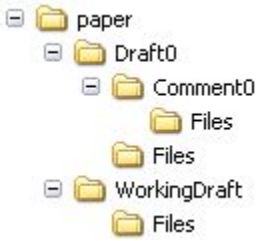

Figure 8 *Directory structure of a Blaise file*

Blaise has an infrastructure set up so that plagiarism detection may be added in a future version. This would include searching papers in the database to see if there are any close matches to a paper that had already been submitted. Although this has not yet been implemented in our current release, there is a form of plagiarism prevention that has been coded into the Blaise Editor: students are not allowed to paste text from other documents

into a Blaise document. In other words, cutting and pasting is allowed within the Blaise Editor, but not between it and anything else.

Related Work

We review related work in this section. André and Richy [1] designed a system for paperless editing of documents that, much like Blaise, is designed for use with a tablet PC or similar input device. However, this system is focused on recognizing gestures input by the proofreader and inserting the corresponding corrections in real-time. Blaise, on the other hand, focuses on leaving markup on the paper as feedback to the student. The instructor may not change the text itself. Also, Blaise is set up both for multiple drafts as well as multiple graders to enable peer-review. If multiple reviewers are grading the paper in Blaise, none will see comments made by any of the other reviewers, and the student may view each reviewer's comments individually. Finally, the system by André and Richy makes no mention of a larger system aimed at automating the process and maintaining long-term portfolios.

Pérez-Quiñones et al [5] describe the use of tablet-PC's to provide feedback in computer science courses. They describe the results of an experiment conducted to evaluate the value of feedback in an object-oriented design course. A system in development at Drexel University [6] is similar to Blaise in that it allows graders to make marks directly on the document, as well as attach notes. That system uses Adobe Acrobat's PDF document format, converting documents if necessary. Acrobat provides a means of marking and adding notes to PDF documents, so the bulk of the system is centered around a web client that uses WebCT to download submissions and convert them to PDF. Blaise, on the other hand, uses its own file format for documents, meaning that no conversion to PDF is necessary. Also, the Blaise file format allows for students to write multiple drafts of the same paper. Another important difference is that while the Drexel system relies on an external client for transmission of papers, Blaise has the transmission components embedded into it, and has its own web client for use on computers that do not have Blaise downloaded onto them. Finally, PDF is a proprietary format and does not come with open-source software. Blaise is both free and open-source, as well as multi-platform. One major purpose of making it open-source is to see whether others are interesting in building upon Blaise to add features or to customize it for their courses.

Summary

This paper described Blaise, a system designed to enable paperless submission and grading of student papers. With its focus on multiple drafts, peer-review and document-tracking on the backend, Blaise is intended to support modern writing programs in which students expect a variety of feedback, must maintain writing portfolios and in which data must be maintained to help monitor the program itself. Blaise has been written in a multi-platform language and is being released (www.blaise.gwu.edu) as open-source to encourage customization and experimentation.

References

[1] Jacques André, Hélène Richy (1999). *Paper-less editing and proofreading of electronic documents*. In EuroTeX '99 Proceedings.

[2] Joanna Bull (1999). *Computer-Assisted Assessment: Impact on Higher Education Institutions*. In Educational Technology & Society 2(3).

[3] Pedro Miguel Coelho, Gonçalo França Faria, David Gonçalves, Joaquim A Jorge. *Using Sketches to Edit Electronic Documents*. http://www.inesc-id.pt/pt/indicadores/Ficheiros/934.pdf

[4] Andrew Ferrier (2004). *Is There a Facilitator on the Phone? The Role of the Teacher in Online Learning*. In Proceedings of the LASTED Educational Conference.

[5] Manuel A. Pérez-Quiñones, Scott Turner. *Using a Tablet-PC to Provide Peer-Review Comments.* http://eprints.cs.vt.edu:8000/archive/00000768/01/Report.pdf

[6] Jeffrey L. Popyack, Nira Herrmann, Bruce Char, Paul Zoski, Chirs Cera, Robert Lass. *Pen-Based Electronic Grading of Online Student Submissions.* http://duplex.cs.drexel.edu/docs/Syllabus_Drexel_Handout.pdf

[7] Mary Queitzsch. *The Northwest Regional Profile: Integration of Technology in Preservice Teacher Education Program.* http://www.netc.org/preservice/challenge.html

[8] Hong Wu, Kåre Sörby. *From Paper Work to a PC World – Online Practice for a Large on-Campus Class*. In Proceedings of the 2004 Informing Science and IT Education Joint Conference.