

Modular Architecture for Pen-Based Digital Ink on Blended Learning Applications

Dan-El Neil Vila Rosado, Margarita Esponda-Argüero, and Raúl Rojas

Abstract—Pen-based technologies can be combined with different kinds of hardware and software to gain pedagogical benefits on blended learning applications. Particularly, digital ink and handwriting recognition furnish the convenience to access, navigate and manipulate data more easily. It is also well known that user interfaces for pen computing can be implemented in several ways, with different techniques and for several kinds of applications. A modular programming architecture for pen-based digital ink was developed in order to provide a robust, reliable, usable and sustainable multimedia technology for a blended learning application. We present this approach and the respective advantages.

Index Terms—Multimedia technologies, blended learning, modular programming, pen computing, sketch recognition.

I. INTRODUCTION

Actually, in several educational organizations, teaching is supported by multimedia technologies at different levels. In these organizations, we can find from just a simple personal computer (PC) until a complete intelligent environment. The gap between traditional and E-learning is narrowing. In this sense, Blended learning emerge like the convergence of online and face-to-face education. Josh Bersin gives a definition of blended learning as the combination of different training “media” (technologies, activities, and types of events) to create an optimum training program for a specific audience [1]. From this point of view, blended learning is a solution to integrate the different kinds of technological advances with the interaction offered in the traditional learning.

Among the technologies to use in a blended learning session we can find blogs, wikis, online tools and online material, e-books, podcast and digital ink.

Inking is the ability to scrawl (i.e., to write or draw awkwardly, hastily, or carelessly) directly on the screen of an electronic device with a stylus [2]. An electronic inking process gives end-users the opportunity of converting the handwritten data to objects that could be analyzed, processed, interpreted or animated, all depends of the use or applicability of this data in a teaching session. Another advantage of digital ink it is that the users (teachers, students or developers) can preserve the data in a raw format and lose none of the respective functionality.

The contribution of digital ink technology to blended learning could be analyzed from three related perspectives:

- Student perspective: Students can take notes in a course,

save their inking to the document, review and search through them later in order to reinforce understanding and improve memory retention.

- Teacher perspective: Instructors on a blended learning session are able to improve their prepared notes (or complete multimedia teaching session) by capturing pointing, drawing, writing, annotating, highlighting, animating that could occurs before, during and after the presentation. In this sense, the teacher can use different teaching methods and incentive interactivity in order to enhance learning.
- Developer perspective: Incorporate digital ink on a blended learning application it depends on the level of objects manipulation that we need to implement. The situation forces to combine different kinds of software design techniques when the main goal is provide a robust, reliable, usable and sustainable system.

Taking account of the perceptions of digital ink technology, students feel engaged when the prepared notes encourage their direct involvement with the class discussion through inking and many researchers embolden to the scientific community to analyze the types of effective, high-quality inking; clarify instructional methods for each type; and creating experimental design studies [2].

Following this motivation, we developed PowerChalk [3] as an interactive media tool that provides an adaptive, modern, flexible, technology-friendly and pedagogical approach suitable to any blended learning environment. Inside of this blended learning application we developed a set of modules to support pen-based digital ink tools. In this paper we propose and describe a modular architecture for several pen-based tools that could be used in a blended learning session for the purpose of enhancing learning and teaching.

This paper is structured as follows. We review the related work in Section II. Then, in Section III we describe the software architecture, and in Section IV the different modules of the system. Finally, in Section V, the implications of our findings and further research are discussed.

II. RELATED WORK

Digital ink applications are proliferating in several domains, for example, educational tools, group decision-making tools, graphical visualizations tools, etc. Some representative examples are:

- 1) NotePals. It is a project developed by UCLA-Xerox [4] as an ink-based, lightweight note sharing system that gives group members easy access to each other's experiences through personal notes on digital assistants.
- 2) K-Sketch. Interface design for creating informal

Manuscript received September 25, 2013; revised November 25, 2013.
The authors are with the Institut für Informatik, Freie Universität Berlin, Takustr. 9, Room 160, CO. 14195, Berlin, Germany (e-mail: vila80@zedat.fu-berlin.de, esponda@inf.fu-berlin.de, rojas@inf.fu-berlin.de).

- animations from sketches. Developed by Univ. of Washington and Univ. Of California [5].
- 3) PADDs. Digital documents that can be annotated either on a computer screen, or on paper through pen-based digital ink. University of Maryland [6].
 - 4) E-Chalk. Electronic chalkboard developed by the Freie Universität Berlin [7].
 - 5) Microsoft OneNote. Computer program for free-form information gathering and multi-user collaboration. It gathers users' notes (handwritten or typed), drawings, screen clippings and audio commentaries [8].
 - 6) Calliflower. Digital ink-based application to view, share and annotate documents easily in a web meeting [9].

The systems above are specialized to a very specific task and have different limitations, among which we mention: cost, hardware or software limitations, inefficient software architecture, they can only work with certain types of data, lack proper software engineering, and present difficulty to evolve or update for developers or end-users.

The use of modular programming like a base structure to digital-ink systems let us to get a sustainable application because resolve the difficulty to evolve or update as long as the capability to work with different kinds of data.

III. SOFTWARE ARCHITECTURE

The digital-ink based modules are built upon the modular application called PowerChalk (Fig. 1). This blended learning application is composed of smaller, separated chunks of code that are well isolated. They can then be developed in separated teams with their own life cycle, and their own schedule. The results can then be assembled together by a separate entity. PowerChalk has a modularized architecture for distributed development based in Java-NetBeans technology [10].



Fig. 1. PowerChalk in a digital tablet.

Such a modular architecture has the followings advantages [11]:

- It simplifies the creation of new features.
- It makes it easy for users to add and remove features.
- It makes it easy to update existing features even to code

level.

- Fast application development.

With these benefits digital ink application becomes a modern, configurable approach to any blended-learning situation, lesson-planning or intelligent environment.

The modular architecture of PowerChalk is shown in Fig. 2.

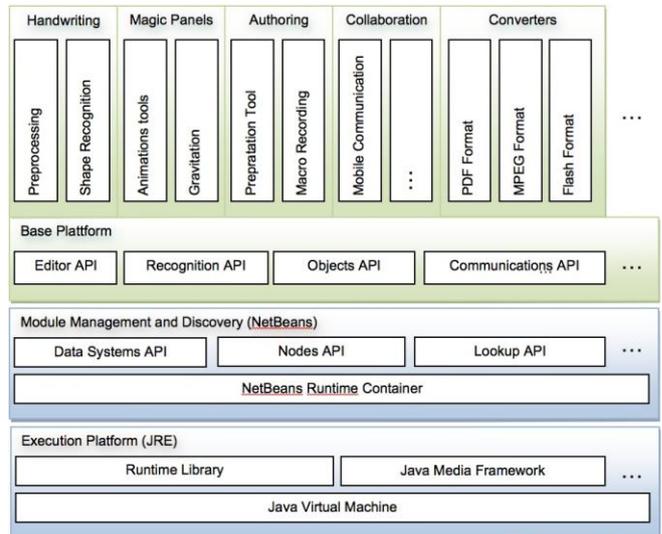


Fig. 2. PowerChalk structure.

IV. MODULES

In the PowerChalk system, the modules work together to improve our experience as end-user. The set of modules that allows to adapt our digital ink structure to any application of a blended learning environment in PowerChalk are the following (Fig. 3):

A. Pre-Processing Module

Data collected from users (strokes) are often incomplete, noisy and inconsistent. Depending on the application and the general purpose, the developer needs to pre-process before the data before any other step like recognition, animation or just final presentation to the user. All the techniques to refine the data are included on this module, for example: sampling, noise reduction (smoothing, filtering, interpolation, wild point correction, dehooking, dot reduction), normalization, affine transformations and stroke connection [12], [13].

B. Shape Recognition Module

This module is for the pattern recognition of shapes of writing units. In this module we can implement shape recognition methods for characters, words, gestures, mathematical notation, line drawings, symbols, schemes, etc.

A lot of techniques for handwriting recognition have been investigated. This module is a set of four general modules (by the moment), each module gather a set of methods based on approaches of pattern recognition: template matching, statistical techniques, structural techniques and neural networks [14].

Actually we have developed only template matching methods. The most commonly used method among our applications (or modules) is a lightweight multistroke recognizer [15].

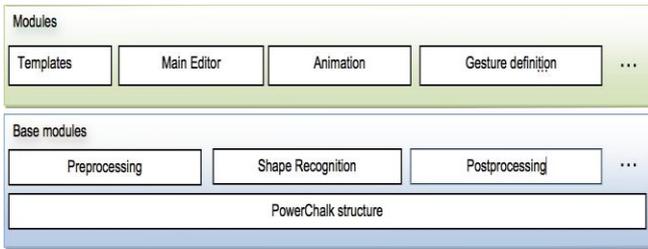


Fig. 3. Digital ink modular architecture.

C. Post-Processing Module

Several shape recognition methods produce alternative output choices, which are desirable for accuracy recognition or a post-processing. In this sense, Post-processing is processing of the output from the preview recognition module [13]. This module could contain different postprocessors:

- Set up a measure of confidence when the recognition output has several alternatives in order to improve automatic recognition or end-user selection.
- String correction algorithms (i.e. When the recognizer has only one output for each character).
- Probabilistic models for insertion, deletion or substitutions in words recognition or mathematical formulations.
- Algorithms based on dictionary lookups or hypothesis generation and test for string correction.
- Algorithms based on higher-level linguistic rules such as syntax and semantics that can increase accuracy in the final recognition.

The actual version of PowerChalk consists on measure of confidence algorithms and string correction algorithms.

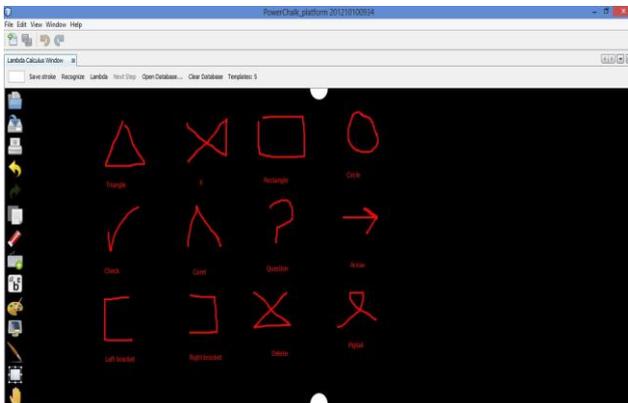


Fig. 4. Templates module.

D. Templates Module

The recognition systems based on template matching are dependent on a set of created templates of their elementary handwriting segments and represent them by the feature parameters used for recognition. In the shape recognition step, these systems compute some distance measure from the data input to the saved templates segments and for each template, they generate a likelihood value depending on the closeness between them [16].

The templates modules encapsulate a simple editor to save, copy, erase and modify the set of templates available for some shape recognition method. It should be noted that this module has communication with the Preprocessing module (Fig. 4).

E. Main Editor Module

The base of any intelligent environment for e-learning is an interactive multimedia area and electronic ink. We have developed an editor area for the PowerChalk canvas where we can add electronic ink components, several multimedia objects (images, keyboard input, etc.) and edit functionalities (Fig. 5). The editor includes pen-based applications to process the different kinds of objects through annotation, correction, condensation, organization, zoom abilities, print options (normal and PDF converter) and creation of slides.

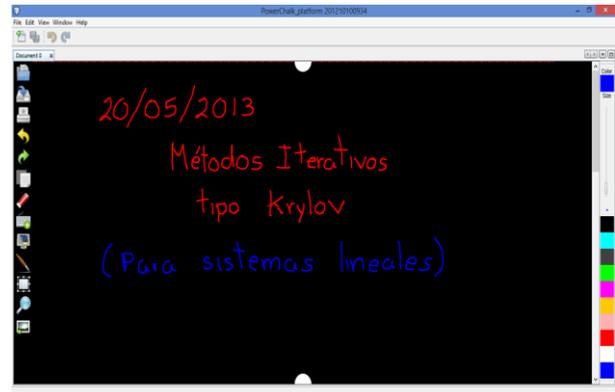


Fig. 5. A view of the main editor.

The main editor has a multiple overlapping layer mechanism to show different levels of multimedia data and allow visualizations, annotation, or underlining.

The following modules (modules F to H) were developed taking as a base the modules above (modules A to G) like evidence of the feasibility to implement new applications and techniques on gesture, handwriting or sketch recognition.

F. Gesture Definition Module

We developed this module to facilitate the incorporation of gestures into user interface prototypes. The recognizer called by this module is the "\$1 recognizer" [17]. This recognizer has limitations as a result of its simplicity, but offers reasonable recognition rates for simple uni-stroke symbols. By the moment, we use 3 unistroke gestures for executing erase, undo and redo commands on the Main Editor canvas. A set of maximum 24 symbols was used on the test phase of this module. The time of development was 2 weeks for a bachelor student.

G. Animation Module

Animation is a powerful mode of communication, but it is out of reach for many users either for a lack of knowledge, time or a portable-efficient tool. The ability to create or use animations in a blended learning session can increase the attention, participation and enhance learning. For this module we follow some requirements and details for informal animation sketching proposed in [18] in order to develop an efficient tool for quick sketching.

To endow PowerChalk with an informal animation tool we merge the structured 2D graphics framework Piccolo2D [19] with an efficient lookup system. The design of the object structure with the lookup system in PowerChalk let us to implement commands to animate objects over the PowerChalk-canvas on the fly, for example, we can animate affine transformations over an object and composite a set of

elements. The result of the animation can be viewed in real time. Development time: 8 weeks.

H. Collaboration Module

A collaboration module for applications is a bridge module between PowerChalk and another software that could process or interpret the PowerChalk data. Examples of software that could collaborate or connect with PowerChalk are: MATLAB, OCTAVE, MATHEMATICA, Gnuplot, etc.

With this kind of modules we can send instructions for plotting functions, expression evaluation, solving equations, running a script, etc. and see the output on the PowerChalk canvas (Fig. 6). We have designed an API for this kind of modules. By the moment we have developed a bridge between PowerChalk and GNU PLOT [20]. The call to Gnuplot, or another application is made through a keyboard input or handwriting strokes in the PowerChalk canvas.

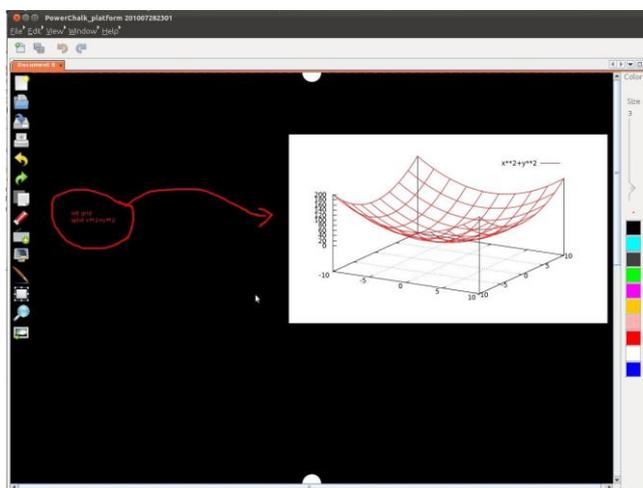


Fig. 6. Collaboration module example (PowerChalk-GNUPlot).

V. CONCLUSIONS AND FUTURE WORK

Supply PowerChalk with a handwriting recognition tool preserves the pedagogical benefits of the traditional chalkboard and provides the possibility to show not only results or isolated ideas but the train of thoughts. Combining the advantages of an interactive multimedia tool and the faculty to be easily updated and adapted on different module applications of PowerChalk, a modular architecture allows it to be a useful and efficient method to embed handwriting recognition into any intelligent environment or any blended learning application.

Test in Windows, Linux and Mac systems were conducted, also with different hardware configurations (tablet-PC, data wall, notebooks), and end-users. Due to the straightforwardness, several end-users (teachers, students and developers) has been involved in the development of new modules: lambda calculus reduction based on handwriting recognition, algorithms animation, Python interpreter based on handwriting recognition, etc.

The consensus is that the modular architecture for the handwriting recognition applications in PowerChalk it is a collaborative, robust, reliable, usable and sustainable interactive multimedia method for blended learning and also let us to develop friendly applications to review and share information on-line or off-line. The modular architecture for

handwriting recognition systems is a reliable method for several intelligent environments or blended learning systems.

It must be pointed out that the modular structure of PowerChalk let us amend quickly any problem in the system particularly in the handwriting system and the data manipulation gained of this process. A complete usability test it is in progress, in order to assess in a measurable way the advantages of modular programming in PowerChalk and the handwriting recognition system. Also, we are increasing the efficiency of every module. For example we are developing:

- New tools for animating objects on the canvas. The objective it is to provide a complete platform for quick sketching to the end-user.
- New Bridge modules to communicate PowerChalk with software like MATHEMATICA, MATLAB, OCTAVE, etc. are under construction.
- New shape recognition algorithms and post-processing techniques.

Because of this advantage, teachers and students have been proposing new PowerChalk modules to design and develop. Among the future modules we can find: Algorithms animations, virtual labs sketched, etc. The idea is to transform the PowerChalk system into a complete intelligent platform for sharing and processing general information and also to show that Modular programming is an efficient method for pen-based digital ink applications.

Important contributions of propose a modular architecture for pen-based digital ink on blended learning applications are: to incentive the creativity among students and teachers to present ideas or train of thoughts in a learning session; improve the design, development operation, efficiency and maintenance of a blended learning application due to the faculty to be easily updated and adapted and finally to involve students, teachers and developers in the development of new learning tools.

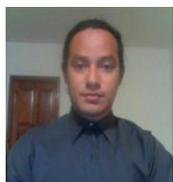
ACKNOWLEDGMENT

We would like to express our gratitude to the students from the Freie Universität Berlin: Alexander Dümont, Christian Tietz and Pawel Osiak. They have implemented and tested some of the handwriting recognition methods as well as developed some of the modularized applications in PowerChalk.

REFERENCES

- [1] J. Bersin, *The Blended Learning Book: Best Practices, Proven Methodologies and Lessons Learned*, San Francisco, California, Pfeiffer, 2004.
- [2] K. Reins, "Digital tablet PCs as new technologies of writing and learning: A survey of perceptions of digital ink technology," *Contemporary Issues in Technology and Teacher Education*, vol. 7, no. 3, pp. 158-177, 2007.
- [3] D. N. V. Rosado, R. Rojas *et al.*, "PowerChalk: An Adaptive e-Learning Application," *Multimedia and Internet Systems: Theory and Practice, Advances in Intelligent Systems and Computing*, vol. 183, 2013, pp. 179-188.
- [4] R. C. Davis *et al.*, "NotePals: Lightweight note sharing by the group for the group," in *Proc. the SIGCHI conference on Human factors in computing systems*, 1999.
- [5] R. C. Davis, B. Colwell *et al.*, "K-sketch: A 'kinetic' Sketch Pad for Novice Animators. Proceedings CHI '08," in *Proc. the SIGCHI Conference on Human Factors in Computing Systems*, ACM New York, NY, 2008, pp. 413-422.
- [6] F. Guimbertiere, "Paper Augmented Digital Documents," *Computer Human Interactions Letter*, vol. 5, issue 2, 2003, ACM, pp. 51-60.

- [7] K. Jantz, G. Friedland, and R. Rojas: "Ubiquitous pointing and drawing," *International Journal on Emerging Technologies for Learning (IJET)*, ISSN: 1863-0383, Wien, Austria 2007
- [8] *Onenote*, Microsoft Inc., 2010
- [9] Calliflower. Iotum Inc., [Online]. Available: <http://www.calliflower.com/>
- [10] Netbeans. [Online]. Available: <http://www.netbeans.org>
- [11] T. Boudreau, J. Tulach, and G. Wielenga, "Rich Client Programming, Plugging into the NetBeans Platform," Prentice Hall, 2006, ISBN: 0-13-23548.
- [12] K. C. Santosh and C. Natte, "A comprehensive survey on online handwriting recognition technology and its real application to the Nepalese natural handwriting," *Kathmandu University Joournal of Science, Engineering and Technology*, vol. 5, no. 1, January, 2009, pp 31-55.
- [13] C. C. Tappert, C. Y. Suen, and T. Wakahara. "The state of the art in on-line handwriting recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 8, August, 1990.
- [14] M. Dzulkifli, M. F. Zafar, and M. R. Othman, "On-Line Cursive Handwriting Recognition: A Survey of Methods and Performances," 2006.
- [15] L. Anthony and J. O. Wobbrock, "A lightweight multistroke recognizer for user interface prototypes," in *Proc. GI '10. Proceeding of Graphics Interface*, 2010, pp. 245-252.
- [16] H. S. M. Beigi, "An overview of handwriting recognition," in *Proc. the 1st Annual Conference on Technological Advancements in Developing Countries*, 1993, Columbia University, pp. 30-46.
- [17] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gesture without libraries, toolkits or training: a 1 dollar recognizer for user interface prototypes," in *Proc. the 20th annual ACM symposium on User interface software and technology*, pp. 159-168, ACM, New York, 2007.
- [18] R. C. Davis and J. A. Landay, "Informal animation sketching: requirements and design," in *Proc. AAAI 2004 Fall Symposium on Making Pen-Based Interaction Intelligent and Natural*, Arlington, VA, October 21-24, 2004, pp. 42-48.
- [19] B. Bederson, J. Grosjean, and J. Meyer, "Toolkit design for interactive structured graphics," *IEEE Transactions on Software Engineering*, 2004.
- [20] Gnuplot. [Online]. Available: <http://www.gnuplot.info/>



Dan-El Neil Vila Rosado was born in Campeche, Campeche, Mexico in 1980. He is a PhD student from Freie Universität Berlin, localized in Berlin, Germany since 2008. He has a bachelor in mathematics from University of Guanajuato (1998-2003) and a Master degree in Computer Sciences and Industrial Mathematics by the Research Center in Mathematics (Guanajuato, Mexico 2004-2006). The author's major fields of study are: Optimization, Soft computing, Artificial Intelligence and Educational Technology.

He was coach in the Regional Mathematics Olympiad Campeche (1998-2000), coworker in the Center for Research in Mathematics A.C. Department of Technology Services (2000-2004). Also, he was software developer in the Consulting and Technology group (CONTECK 2005-2007) and project coordinator (2007-2008). He worked like teacher, research assistant and head of division in the University of Guanajuato (2002-2007) and Institute Technology of Irapuato (2007-2008).

M.Sc. Vila has published refereed conference papers supporting his work on robotics, optimization, artificial intelligence and E-learning.



Margarita Esponda-Argüero is a professor of algorithms and programming languages at Freie Universität Berlin since 2009. Previously, she held professorships at Technical University of Gießen-Friedberg and at the Technical University of Wildau.

Margarita Esponda obtained his bachelor degree in Mathematics and Physics from the National Polytechnic Institute of Mexico and the master degree in Informatics from the Technical University of Berlin. She received a PhD in Informatics from FU Berlin in 2004.

Prof. Esponda has been working on techniques for visualizing data structures in algorithmic animation systems, electronic voting systems and multimedia technologies. She has published refereed conference and journal papers supporting his work on educational computing research, history of computing and information visualization.



Raúl Rojas is a professor of Artificial Intelligence at Freie Universität Berlin since 1997. He received a PhD in Economics from FU Berlin in 1989, where he also obtained his Habilitation degree in Computer Science in 1994.

Prof. Rojas has been active in the field of artificial intelligence since 1986, having been a member of the Prolog Machine team at GMD-First (now Fraunhofer

Prof. Rojas has been active in the field of Artificial Intelligence since 1986, having been a member of the Prolog Machine team at GMD-First (now Fraunhofer Gesellschaft). He has published extensively about neural networks, robotics, intelligent systems and the history of computing.

Rojas is a Member of the Mexican Academy of Sciences. He received the Technology Transfer Prize 2008 from the Berlin Technology Foundation and the Gold Medal Heberto Castillo from Mexico City for his contributions to robotics.