

Improving e-Learning Environments for Pen and Multi-touch Based Interaction

A study case on blog tools and mobile devices

André Constantino da Silva
Institute of Computing (PG)
UNICAMP, IFSP
Campinas, Brazil, Hortolândia, Brazil
acsilva@ic.unicamp.br

Heloísa Vieira da Rocha
Institute of Computing, NIED
UNICAMP
Campinas, Brazil
heloisa@ic.unicamp.br.br

Abstract— e-Learning environments are applications that use the Web infra-structure to support teaching and learning activities; they are designed to have good usability using a desktop computer with keyboard, mouse and high resolution medium-size display. Devices equipped with pen and touch sensitive screen have enough computational power to render Web pages and allow users to navigate through the e-learning environments. But, pen-based or touch sensitive devices have a different input style; decreasing the usability of e-learning environments due the interaction modality change. To work on mobile contexts, e-learning environments must be improved to consider the interaction through pen and touch. In a previous work, we presented the InkBlog, a blog tool that receives input data from pen allowing users to handwrite posts. In this work, we present an extension of InkBlog to receive input from multi-touch screens. We described the changes over the InkBlog architecture and implemented components to treat data generated by pen and touch.

Keywords— Human-Computer Interaction; Electronic Learning Environment; Mobile Devices; Interaction Styles.

I. INTRODUCTION

e-Learning environments, such as Moodle [1], SAKAI [2], TelEduc [3], Ae [4], are applications that use the Web infra-structure to support teaching and learning activities. The e-Learning environments are designed to support a variety of users and learning contexts, but they are designed to conventional computers, usually equipped with keyboard and mouse as input and a medium screen and speakers as output; a limited set of interaction styles for nowadays devices. These modalities, and the technology that support them, shape the teaching and learning activities done in the e-Learning environments; they focus on reading and writing skills.

One example of tools available at e-learning environments is Weblog, a communication and collaborative tool that aims to promote the sharing of messages among participants through an area named blog. Users can publish texts, images, audio, videos and links, sharing their opinions, in posts typically displayed in reverse chronological order (the most recent post appears first) and allowing visitors to leave comments. In this way, blogging can be seen as a form of social networking.

Mobile devices, such as smartphones and tablets, are becoming increasingly popular; most of them have touch screen displays, Internet access and enough computing

power to process Web pages. So, it is possible to access blog tools, read the messages, post new messages and write comments through mobile devices. But, it is important to consider that these tools (and so their Web pages) are developed to be accessed by desktop computers equipped with keyboard, mouse and a medium size display; in your previous work we described that when a user interface designed for a set of interaction styles is accessed by a different set of interaction styles the users face interaction problems [5]. Another problem is that it is not possible to take advantage of the interaction style features; for example, in a desktop computer, users use the keyboard to typing the post text. In a pen-based computer without handwriting recognition, users need to type each letter pressing the pen in the respective key in a virtual keyboard. This way of writing texts takes a lot of time, make boring the writing task and do not take the mainly pen purpose: handwriting and do sketches easily. In the case of touch sensitive screen, the user can touch in the virtual keyboard to write the post, but it is not possible to do sketches.

So, we believe that blog tools and e-learning environments need to be improved to be easier to use in some devices and some contexts, e.g., areas which need sketches or drawing such Mathematics. In our previous work [6], we developed the InkBlog tool to easily write handwrite or sketched posts in pen-based devices by adding features to manipulate electronic ink into a blog tool from Ae, an e-Learning environment. But, the InkBlog have some limitations: display handwriting posts correctly in smartphones, but it is not possible to sketch post. In this work, we add some features to improve the InkBlog to allow users to write messages in many devices: desktop computers, smartphones and tablets.

Section II presents a literature review about electronic ink technology and e-Learning environments, focusing on Weblog tool. Section III presents the InkBlog, describing how the technologies are employed to allow users handwriting posts and comments using pen-based devices. In Section IV, we present the modifications done in InkBlog to allow users to interact using multiple fingers. The last section presents the conclusion and future work.

II. E-LEARNING ENVIRONMENTS AND INTERACTION STYLES

The World Wide Web has changed since its invention from a static to a highly dynamic media in the recent years; so, the term “Web 2.0” was coined in 1999 to describe the

web sites that use technology beyond the static pages and its uses for collaborative, user-centric content production and interactive content access [7]. Safran, Helic and Gütl [8] describe that in literature the marks of Web 2.0 include: (i) social phenomena, such as the Web for participation, (ii) technology for significant change in web usage, and (iii) design guidelines for loosely coupled services. The Web 2.0 allows users to interact and collaborate with each other in social networking sites, weblogs, podcasts, wikis, video sharing sites and other sort of tools.

One kind of Web applications that have some Web 2.0 features is e-Learning environments, as Moodle [1], SAKAI [2] and Ae [3]. They are applications with tools to support teaching and learning activities through the Web. Tools in these environments allow users to create content, communicate with other users and manage the virtual space. Tools like chat, forums, portfolios, repositories are widely used, and tools that explore the audio and video resource to user communication, such as instant messenger and video-conferences, are becoming common among the environments.

HyperText Markup Language (HTML) is used for any web application to describe the page interface and its content. Usually, in web applications that users post text, there is a rich text editor to allow users write formatted text without HTML skills. In desktop computers, the users use the keyboard to typewrite the letters, and use the mouse to select format functionalities (some of them have shortcuts to be triggered by the keyboard). Since the rich text editors have a direct manipulation interface similar as text editor applications, it is easy to be used in desktop computers equipped with mouse and keyboard.

The HTML have some improvement defined in the last version, the HTML5, related with support multimedia, keep it easily readable by humans and consistently understood by computers and devices [9]. HTML5 adds the new `<video>`, `<audio>` and `<canvas>` tag elements, as well as the integration of Scalable Vector Graphics (SVG, a vector image format for two-dimensional graphics based on eXtended Markup Language - XML) content and Mathematical Markup Language (MathML, a XML based-format to describing mathematical notations) to integrate mathematical formulae into Web pages. These features are designed to easily include and handle multimedia and graphical content on the web without having proprietary plugins and Application Programming Interface (APIs) installed.

The `<canvas>` tag allows for dynamic, scriptable rendering of 2D shapes and bitmap images; it is a drawable region defined in HTML code with height and width attributes. JavaScript code may access the area through a full set of drawing functions similar to those of other common 2D APIs, thus allowing for dynamically generated graphics.

Another evolution in HTML is standardizing how the browser must handle events from touch and pointer inputs [10]. The W3C Group specified that “The Touch Events specification defines a set of low-level events that represent one or more points of contact with a touch-sensitive surface, and changes of those points with respect to the surface and

any DOM (Document Object Model) elements displayed upon it (e.g., for touch screens) or associated with it (e.g., for drawing tablets without displays)”. This specification was done thinking in devices equipped with stylus such as tablet and define event types for: (i) when a user touch the surface (touchstart), (ii) when a user remove a touch point from the surface (touchend), (iii) when a user moves a touch point along the surface (touchmove), (iv) to indicate a touch point has been disrupted (touchcancel). Having different event types for input data generated by each modality gives flexibility for the developers define the actions to be trigger for each modality.

W3C defines XML formats for non-primitive data to allow exchange of a wide variety of data on the Web and elsewhere; one example is Ink Markup Language (InkML) [11]. The InkML provides a common format for exchange ink data between components such as handwriting and gesture recognizers, signature verifiers, sketches, music and other notational languages in applications. The InkML serves as data format for represent ink gathered by an electronic pen or stylus. It is possible to find some uses of InkML, such in Microsoft Word 2010 support electronic ink in text review and the InkML JavaScript Library [12], that offers some functions to allows InkML digital ink to be referenced within Web pages and rendered directly into the HTML5 `<canvas>` tag.

Considering the technology breakthrough that HTML5 proposes, most of the web sites use HTML5 to impress users through content exhibition; few developers take care about the user input interaction styles, so they develop web pages for users with keyboard and mouse in desktop computers and not appropriate for touch devices. But, this scenario is changing with the smartphone and tablet popularization: the web designers need to think about the other interaction styles, such as touchscreen and pen-sensitive devices.

In pen-based devices when the user moves the pen in the screen, the pen trace should result in electronic ink that must be treated by the application to be rendered and stored. But, desktop applications, that running in the Tablet PCs, do not treat electronic ink, so it is necessary to incorporate special applications to treat the electronic ink to have benefices of the pen interaction style. This is the case of Classroom Presenter [13] and others pen-based software. In our previous work [14] we studied three applications (Professional Adobe Acrobat, Windows Journal and Jarnal) to do classes annotations or writing manuscript texts in student activities using a Tablet PC, describing some identified usability problems. We conclude that desktop applications that do not manipulate the electronic ink loose the Tablet PC potential because the pen is used just as a pointer device. This is valid for web applications.

In our previous work, trying to develop web applications that do a better use of pen interaction, we developed the InkBlog for Ae environment, described in the next Section. We chose a blog tool because edublogs, blogs with educational purpose, archiving and supporting student and teacher learning by facilitating reflection, questioning by self and others, collaboration [15] and by providing contexts for engaging in higher-order thinking. Ray [15] cites four ways

to incorporate blogs into the classroom, including: (i) using them to communicate information to students and parents, (ii) to provide instructional resources and useful links, (iii) to allow students the opportunity to collaborate with one another on various projects without being in the classroom itself, and (iv) to showcase student work and projects, like poetry and photographs of project work. Another motivation is that users can share draws and images, and the stylus allows users to draw in the device.

III. THE AE ENVIRONMENT AND INKBLOG TOOL

The TIDIA-Ae Project (TIDIA-Ae is the acronym for “Tecnologia da Informação para o Desenvolvimento da Internet Avançada – Aprendizado Eletrônico”, in English “Information Technology for Development of Advanced Internet – Electronic Learning”) was initiated by FAPESP (the State of São Paulo Research Foundation) with the main goal of developing an e-Learning environment that can explore the potential of Advanced Internet and can provide support to different educational context needs: the Ae environment. Several tools are developed for Ae environment such as Portfolio, Mail, Discussion Forum, Chat and Weblog.

In e-Learning environments, each course participant (student or teacher) owns a blog, where she can post or exclude messages sharing texts, video or audio messages with other participants. Each participant may access other course participants’ blogs, comment posted messages by the owner of the blog.

To publish a message in the Ae’s Weblog, the participant clicks in the link “Compose a new post” in her blog page. A new page within a form to be filled will be displayed. After filling up the form, the user clicks on the “Confirm” button and the system will save the new post and redirect the user to

the user’s blog page displaying the new message at the top.

The Weblog uses a rich text editor to allow users to easily write formatted text without having knowledge of HTML. For desktop computers the rich text editor works well, the user type the text in the keyboard and use the mouse or shortcuts to trigger format features. But, in the case of pen-based tablets the usability decreases: the user needs to typewrite each letter pressing the pen over the respective key in a virtual keyboard. It is not possible to handwriting or sketching, actions done with a pen and paper.

Ae’s Weblog tool was designed in a design-evaluation process. The first step to create the Weblog’s user interface was to develop a high-fidelity prototype considering the user will interact in a desktop computer equipped with mouse, keyboard and a high-resolution medium size display. This user interface was evaluated and some modifications were done before codifying phase. A usability test was done with the first implement version, and some modifications (e.g., removing some functionalities and changing some labels) were recommended. The final version is presented in Figure 1.

In our previous work, we presented the InkBlog [6] (Fig. 1). It was created to make easier to handwrite posts and comments using a stylus in pen-based devices. The approach to develop the InkBlog was to increase the Weblog tool with components to generate and manipulate the electronic ink in the user interface, representing the electronic ink in InkML format; before a usability test was done to identify problems when user interacts with pen. Changes in the Weblog’s architecture (Fig. 2) and user interface (Fig. 3) were done to support input data from stylus. In the architecture we added a component to receive data from pen, the InkController component, and a component to renderer this data as electronic ink, the InkRenderer component. Both

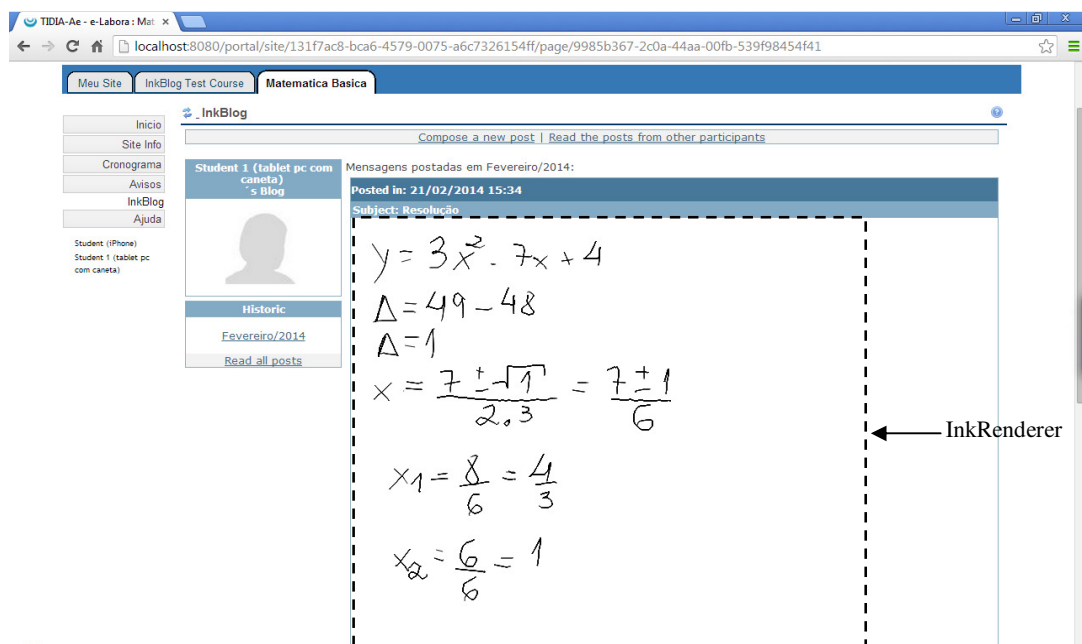


Figure 1. Using InkBlog to share a resolution about finding root for quadratic equation rendered by Google Chrome browser in a Tablet PC.

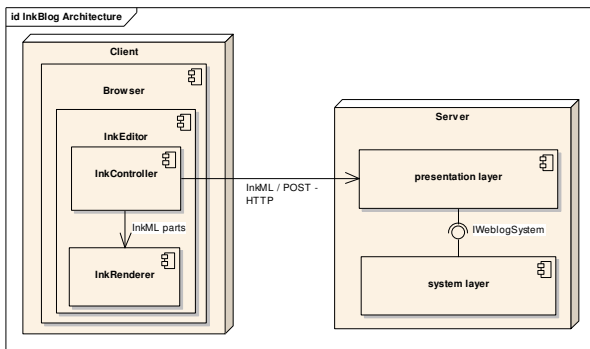


Figure 2. InkBlog components to treat input data from pen.

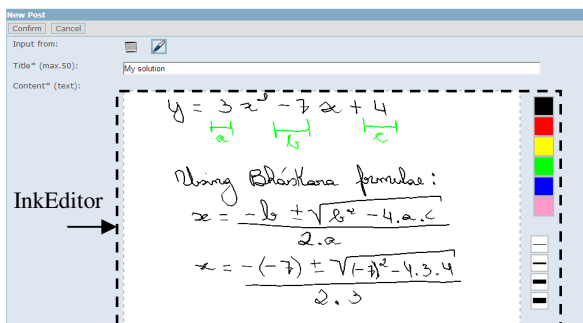


Figure 3. Using InkBlog to handwrite a post using a stylus.

components, InkController and InkRenderer, compose the InkEditor: a handwritten text editor for web pages that render the electronic ink and receive the input data generated by stylus.

The pen input data is received by the InkController, which transforms each point of the trace into coordinate points following InkML format. The user can choose the trace's color and the width selecting the buttons options in the right side (Fig. 3). When the user points out and presses the pen into a color or width button, the next traces will have the brush attributes set to look like the selected options.

The InkRenderer, the other InkEditor component, draws the traces of a handwritten post in the user screen (Fig. 1). The InkRenderer's code, the electronic ink data in InkML format and the HTML page are sent to the client over the HTTP protocol (Fig. 2) to display the page with posts. After all the data and code arrived in the client, the InkRenderer reads the InkML data resident inner the tag canvas, and draws the electronic ink for each trace, taking into account the ink formatting. The InkRenderer was developed using the InkML JavaScript Library.

To post a new message, the user can choose the input hardware (keyboard or pen) selecting the icon on "input from:" field, to type the text using a keyboard or to handwrite a post with a stylus (Fig. 3). When the user chooses the pen, she will write a handwriting post, the browser will hidden the text editor and show the InkEditor, where the user will use the stylus to handwrite. When the user touches the InkEditor within the pen and draws a trace, the InkController will listen to the user actions, getting the dots that compose the trace. Each dot is recorded and a line

connecting the preceding point to the new point is drawn until the user releases the pen. After the pen is released, the InkController will generate the InkML's trace node for the new trace. The user can draw as many traces she wants, all them will be stored and will compose the InkML data. When finished handwriting the post, the user will click in the Confirm button and the generated InkML data will be sent to the server to be stored.

Some changes were needed in the Web application to distinguish textual content from typewriting content and to show the correct editor in the post view. The changes are done in the presentation layer; the other layers have not been changed.

The client device needs to have a compatible HTML5 browser to run the InkEditor. The InkEditor uses InkML to represent the handwriting data and the Canvas HTML attribute to draw the traces in the screen.

It is also possible to handwrite comments and post them: the process is similar to the described process above.

IV. IMPROVING BLOG TOOL FOR PEN AND TOUCH INTERACTION

To support touch interaction, we developed a new version of the InkBlog, extending it with features to manipulate data from touchscreens. To include touch interaction, we adopted the same evolutionary approach as when we included pen interaction: usability test with user interaction with pen (two devices were considered, a smartphone and a tablet) and modifications of the user interface, following usability tests.

Multi-touch in web applications is more common on games; Johnson [16] presents a tutorial to include features of multi-touch in web applications. Analyzing the InkBlog code, we modified the InkController code to support touch devices including the handling of touch events defined by W3C. The first modification was handling the touchstart, touchmove and touchend event types. Since we wanted the users draw with their fingers in touch sensitive devices, these events types call functions to start a line, to compose the line, and to stop to draw a line, respectively. This modification allows users to interact with one finger per time.

To allow multi-touch, it was necessary to use a `<inkml:trace>` to store the data from each finger in a array. The browser send to the function that will handle the user interaction an event object with the `changedTouches` attribute, a collection with data from one or more modified touch points. To identify finger's move it is possible to use the event's `identifier` attribute; this value was used as index in the array of `<inkml:trace>` to put the data in the correct line.

To avoid the browser to scroll the page when the user moves the fingers on the screen, it was called the event's functions `preventManipulation()` and `preventDefault()`.

To test the new version of InkBlog we used a Tablet PC, a tablet and two smartphones. The Tablet PC is a HP TouchSmart TX2-1040br, 2.2 GHz dual-core processor with 3 GB RAM and a 12" multi-touch screen with Windows Vista operating system and with Windows 8 operating system. This model has the design similar to HP laptops but

it is equipped with the described hardware for Tablet PC, for our purposes, a pen sensitive and touch sensitive screen. Google Chrome browser version 31.0.1650.57m was used to navigate through the e-learning environment. So, the used Tablet PC has pen and multi-touch modalities.

The tablet used in the session tests is a Brazilian Positivo Ypy AB10EC, 1.0 GHz ARM Cortex A9 processor with 1 GB RAM and a 10" multi-touch screen 800 x480 pixels of resolution with Android version 4.1.1 operating system. In the iPhone, the used browser was the Google Chrome 31.0.1650.59.

The smartphones used in the session tests are an iPhone 3GS and a Motorola Milestone. The iPhone 3GS has a 833 MHz Samsung S5L8920 ARM Cortex-A8 processor with 256 MB RAM and a 3.5" multi-touch screen 480 x320 pixels of resolution and iOS version 6.1.3 as operating system. In the iPhone, the used browser was the Safari. The Motorola Milestone has a 600 MHz Cortex-A8 processor with 256 MB RAM and a 3.7" multi-touch display 854 x 480 pixels of resolution and Android version 4.0.3 as operation system. To browse in the Web application, the Android stock Web browser was used.

In the post message task, the new version of InkBlog demonstrated to have good performance in all devices. Fig. 4

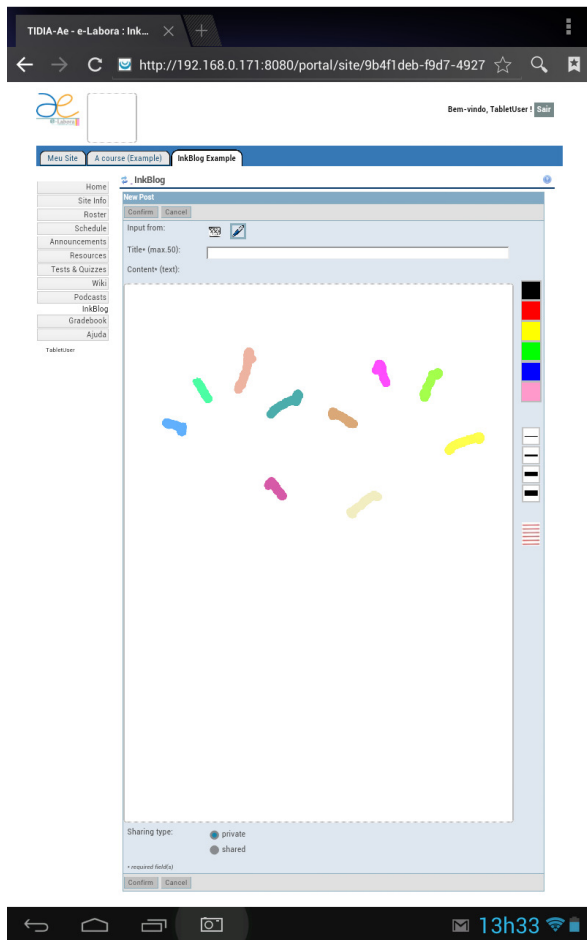


Figure 4. Using InkBlog to write a post using fingers in an android-based tablet in a landscape position.

shows the use of the Positivo Tablet to draw in the InkBlog using ten fingers at the same time (each line was done by one finger, draw here with different colors but the InkBlog just allow user to draw with one). The HP Tablet PC with Windows 8 recognized 4 fingers, instead of 5 fingers described in the operation system user manual. The Motorola Milestone and iPhone (Fig. 5) recognized 5 fingers.

All devices displayed the posts correctly (Fig. 6), but when the number of drawing posts increases, more time is necessary to render. The render process depends on the hardware, so different devices take different time to complete the task.

In all devices, smartphones or tablets, the touch input data was correctly identified by the web application. In Windows 8 using Google Chrome, the input data from the pen is recognized as mouse input.

To manipulate the iPhone and the Android-based smartphone we used fingers and a capacitive stylus. Since these devices cannot distinguish touches by fingers from touches by stylus, it is not possible to handling the input data considering the data origin. In this case, we prefer to allow users drawing on the surface instead of using the data to scroll the screen. We decided considering the user goal: to post a message.

Another handling is possible: to combine the power of input modalities, e.g., in Tablet PCs, devices that have capacity to distinguish the origin of the input data, it is possible to use the data from the pen to generate the electronic ink and the data from touch to scroll the screen or to trigger another gesture such as selection and zoom.

We did one study case based on user test (following Human-Computer Interaction methodology) planned to apply a profile questionnaire, the performance of four tasks and, in the end, apply a opinion questionnaire. In the opinion questionnaire we adopted the Likert scale for multiple choice

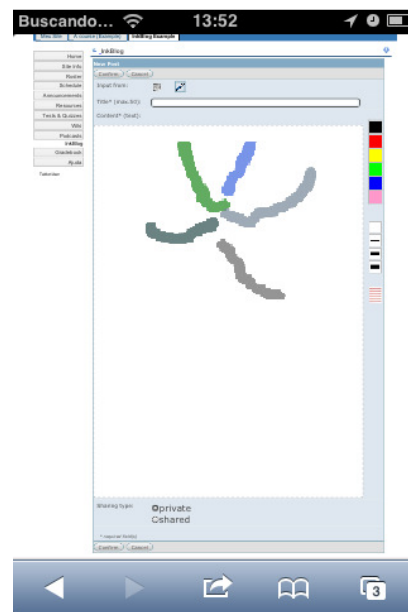


Figure 5. Using InkBlog to write a post using fingers in an iPhone smartphone in a landscape position.

questions. We invited three volunteers: one woman and two men. One of them have post-graduation (33 years old) and the other two are undergraduate student in Computer Science (22 and 27 years old). All of volunteers consider themselves experts and use touchscreen devices every day for more than one year. None of the volunteers used pen-sensitive devices. All volunteers access Internet, one of them for more than ten hours per day and two of them uses between five to ten hours per day.

About e-learning environments, all volunteers have used: one of them does not know the name of the used environment and the other two had used TelEduc and Moodle. Two of them reported that they used for more than one year, and the another volunteer reported that she used between one to five months. Two users reported that they are experts about e-learning environment and another one reported have intermediate knowledge.

The volunteers were questioned about how to help a colleague that asked for help by e-mail to resolve an exercise that needs to apply the Bháskara formulae. One volunteer used special character to represent the power and the square in the formulae. Another one said that she probably sent a link with the formulae and some resolved exercises. The last volunteers answered that probably she will search on the Internet to remember the Bháskara formulae.

All volunteers performed four tasks: i) login (with a given username and password) in the Ae environment; ii) post a message in the InkBlog with the resolution of an exercise that need to find the roots of a quadratic equation; iii) aim a colleague to find errors in her resolution; iv) logout of the environment. The volunteers are asked to perform the four tasks in three devices: i) the Tablet PC using the pen; ii) the iPhone using fingers; and iii) the Positivo tablet using fingers.

By observation, it was possible to identify several problems. The volunteers had some difficult to do small traces (e.g., - and =); this traces are done quickly, and most of time, the interval between the start and the end is not long enough to generate the data; sometimes, the trace is recognized by the browser as gesture (e.g., back or forward). Dots are considered by all volunteers difficult to be done; since they are small, the user needs to press and move slowly the pen (or the finger) in circular move to do a dot. Maybe, these are the reason for the volunteers answered in the questionnaire, when questioned "It was easy write a message using a pen", one user marked the option "strongly agree", another one marked the option "agree", and the last one marked "disagree". One volunteer said that it was difficult to write as legibly as when she writes on the paper.

In the Tablet PC using the pen, the volunteers tried to scroll the page touching the pen in a white area (similar as they did when interact in smartphones). But, this feature is not implemented in the environment and the Chrome browser either. Maybe, this is the reason a volunteers mark "agree" when asked about if her had difficult to scroll the page. The volunteers reported that had some problems to use the virtual keyboard due the small keys.

About the interaction using the iPhone, the volunteers just used one finger to write in the InkEditor. The volunteers

used two fingers only to do gesture for zooming. The

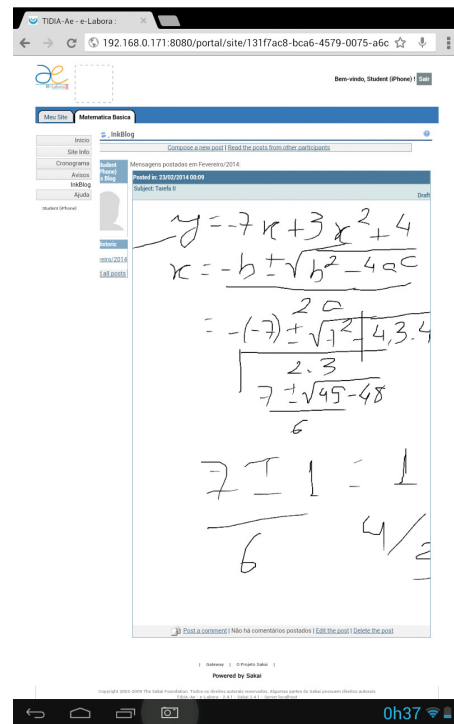


Figure 6. A resolution made by one volunteer with her finger about finding root for a quadratic equation rendered by Google Chrome browser in the Positivo tablet.

volunteers described some difficulty to read the screen, but they did not have difficult to trigger links and icons. One volunteer reported difficulty to scroll the page. About writing the resolution using the finger (question "it was easy write my message through the finger"), a volunteer marked "strongly agree", another volunteer marked "agree" the last one marked "strongly disagree". It is important to highlight that the exercise is easier to be done using a pen, due the pen characteristics. In the open question, one volunteer said "the iPhone screen is too small, and my fingers are large turn hard to do precise strokes and that fit in the available space" (Fig. 6). One volunteer reported the impossibility to do the gesture for zooming in the InkEditor component (InkEditor recognize as electronic ink).

About the interaction using the Positivo tablet, the volunteers reported that they did not have difficult to trigger links and icons, but one volunteer reported had some difficult to read the screen. No volunteer had difficulty to scroll the page (since a touch in anywhere in the screen can generate a scroll). About writing the message within a finger in the Positivo tablet (question "It was easy write my message using a finger"), one user reported "strongly agree", another one reported "disagree", and the last one "strongly disagree". One volunteer reported (and it was observed too) that, after each trace done, the visual focus returned to the field text, and the browser scroll up the page, causing some risks in unwanted places in the InkEditor due the user started the next trace.

V. FINAL CONSIDERATIONS AND FUTURE WORK

e-Learning environments are applications that use the Web infra-structure to support teaching and learning activities. To post text, there is a rich text editor to allow users writing formatted text without HTML (HyperText Markup Language) skills. This solution has good usability on desktop computers, but when the user interacts with a pen or touching, she needs to type each letter using a virtual keyboard, so the usability, most specifically, the efficiency, decreases and makes the writing task boring. Another problem is the difficulty to draw sketches using the mouse.

In our previous work we improved the Ae's Weblog with components to generate and manipulate electronic ink, allowing users handwrite posts and draw sketches in pen-based devices, calling this new tool InkBlog (available to test at [17]). For the InkBlog development, we chose well-defined and promising technologies, such as W3C InkML and HTML5, allowing to codify a light electronic ink editor for Web pages, the InkEditor. Some modifications are done in the Weblog for distinguishing the correct editor to be displayed in the new post composite page and display handwriting posts and comments. However, InkBlog has some limitations: it is not possible, in touchscreen device, to use fingers to draw.

We believe that with InkBlog it is possible to have a better support to disciplines like Graph Theory and Computer Theory. Without the InkBlog, the user needs to use paper and pencil to resolve an exercise and use specific hardware, such a scanner, to digitalize the solution. Or the user needs to use a special application to draw a graph. In both solutions, the user posts the picture as an attached file in a weblog post. Using the InkBlog the user can sketch the graph directly on the weblog tool through direct manipulation.

In this work, we modified InkBlog to handling touch events, allowing users with multi-touchscreen devices post messages with draw done by fingers. The new version of InkBlog was tested in 4 devices (HP Tablet PC, Positivo tablet, Motorola Milestone smartphone and iPhone) and in 3 operation system (Windows Vista and 8, iOS and Android). Usability test was done with three volunteers. Some usability problems were found and they will be corrected in the next version. For future works, we will: 1) do a performance evaluation and evaluated in a class; 2) improve other tools by adding the developed components; 3) support gestures to trigger functionalities by the integration with the environment with a gesture recognizer, and study the gesture's drawing for each function; 4) integrate with a handwriting recognizer to recognize the word and allow a better integration with another environment tools, such as search tools; 5) using a post or a document as background, allowing teachers or other student to do annotations about a post.

ACKNOWLEDGMENT

The authors thank the CAPES and CNPq for grants and for FAPESP through TIDIA-Ae Project (n. 05/60572-1), which provided the equipment.

REFERENCES

- [1] Moodle Trust, "Moodle.org: open-source community-based tools for learning," Available at <<http://moodle.org>>. [retrieved: Jul. 2013]
- [2] SAKAI Environment, "Sakai Project | collaboration and learning - for educators by educators," available at <<http://sakaiproject.org>>. [retrieved: Jul. 2013]
- [3] TelEduc Environment, "TelEduc Ensino à Distância," available at <<http://www.teleduc.org.br>>. [retrieved: Jul. 2013]
- [4] Ae Project. "Ae - Aprendizado Eletrônico Environment," available at <<http://tidia-ae.iv.org.br/>>. [retrieved: Nov. 2013]
- [5] A. C. da Silva, F. M. P. Freire, and H. V. da Rocha "Identifying Cross-Platform and Cross-Modality Interaction Problems in e-Learning Environments," Proc. of 6th International Conference on Advances in Computer-Human Interactions (ACHI 2013), February 2013, pp. 243-249.
- [6] A.C. da Silva and H.V. da Rocha, "InkBlog: A Pen-Based Blog Tool for e-Learning Environments," in Bridging Disciplinary Boundaries: Issues in Informing Science and Information Technology, vol. 10, May 2013, pp. 121-135. ISBN: 1932886699.
- [7] O'Reilly, T. "What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software," available at <<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>>. [retrieved: Nov. 2013]
- [8] C. Safran, D. Helic, and C. Gütl., "E-Learning practices and Web 2.0," Proc. of International Conference on Interactive Collaborative Learning (ICL 2007), Sep. 2007, 8p.
- [9] R. Berjon, T. Leithead, E. D. Navara, E. O'Connor, and S. Pfeiffer, "HTML5 - A vocabulary and associated APIs for HTML and XHTML W3C Candidate Recommendation," available at <<http://www.w3.org/TR/html5/>>. [retrieved: Nov. 2013]
- [10] D. Schepers, S. Moon, M. Brubeck, and A. Barstow, "Touch Events" available at <<http://www.w3.org/TR/touch-events/>>. [retrieved: Nov. 2013]
- [11] Y. Chee, K. Franke, M. Froumentin, S. Madhvanath, J. Magaña, G. Pakosz, G. Russell, M. Selvaraj, G. Seni, C. Tremblay, and L. Yaeger "Ink Markup Language (InkML) W3C Recommendation," available at <<http://www.w3.org/TR/InkML/>> [retrieved: Nov. 2013].
- [12] T. Underhill. "InkML JavaScript Library," available at <<http://inkml.codeplex.com/>>. [retrieved: Nov. 2013]
- [13] R. Anderson, R. Anderson, P. Davis, N. Linnell, C. Prince, V. Razmov, and F. Videon, "Classroom Presenter: Enhancing Interactive Education with Digital Ink". Computer, vol. 40, September 2007, pp. 56-61. doi:10.1109/MC.2007.307.
- [14] A.C. da Silva, and H.V. da Rocha, "Experiences in use Tablet PC to Support Student's Activities: Five Years of an Exploratory Study," Proc. of 4th International Conference on Mobile, Hybrid, and On-line Learning (eL&mL 2012), pp. 38-43.
- [15] J. Ray, "Welcome to the blogosphere: the educational use of blog", Kappa delta pi record, v. 42, n. 4, 2006, pp. 175-177.
- [16] T. Johson, "Handling Multi-touch and Mouse Input in All Browsers - IEblog - Site Home - MSDN Blogs," available at <<http://blogs.msdn.com/b/ie/archive/2011/10/19/handling-multi-touch-and-mouse-input-in-all-browsers.aspx>> [retrieved: Nov. 2013].
- [17] Ae Environment with InkBlog. "Ae - Aprendizado Eletrônico Environment," available at <<http://tidia-ae.nied.unicamp.br/portal>>. [retrieved: Mar. 2013]