Towards an Architecture to Multimodal Tools for e-Learning Environments

André Constantino da Silva
IFSP, NIED/UNICAMP
Hortolândia, Brazil
e-mail: andre.constantino@ifsp.edu.br

Fernanda M. P. Freire, Flávia L. Arantes
NIED/UNICAMP
Campinas, Brazil
e-mail: ffreire@unicamp.br, farantes@unicamp.br

***Abstract*— e-Learning environments are applications that use the Web infrastructure 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 our previous work, we presented InkBlog, Multimodal Editor, and InkAnnotation: three multimodal tools for e-learning environments. Based on these previous works, we propose a generic architecture for multimodal tools for e-learning environments, describing common components to treat data generated by pen and touch that can be generalized to treat other modalities.**

Keywords- architecture for multimodal tools; multiple platform and multidevices; e-Learning environments.

I. Introduction

e-Learning environments, such as Moodle [1], SAKAI [2], TelEduc [3], Ae [4], are applications that use the Web infrastructure 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 for 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.

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 Web applications to read and to post content through mobile devices. But, it is important to consider that most of e-Learning tools are developed to be accessed by desktop computers equipped with keyboard, mouse and a medium size display; in our 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 for typing the content that will be posted. In a pen-based computer without handwrite recognition, users need to type each letter pressing the pen on the respective key of a virtual keyboard. This way of writing text takes a lot of time, makes the writing task boring and does not take advantage of the main purpose of the pen, namely, handwriting and doing sketches easily. In the case of touch sensitive screen, the user can touch the virtual keyboard to write the post, but it is not possible to do sketches.

So, we believe that the e-learning environments and tools that compose them need to be improved to be easier to use in a variety of devices and contexts, e.g., areas which need sketches or drawing, such mathematics; or the environment must be sensitive about the device the user is using or the user's location. So, our research group is developing some tools that take advantage of the interaction styles available on the user device. An obstacle are the few papers already published on multimodal architectures for Web applications, in particular e-learning tools, and lack of models. So, we propose a generic architecture for multimodal tools of e-learning environments.

Section II presents a literature review about e-learning environments, multimodality and multimodal systems. Section III presents our previous work about development of multimodal tools for e-learning environments. In Section IV, we propose a generic architecture for tools that compose e-Learning environments and need to manipulate multimodal inputs, allowing this kind of system to be more adaptable to the context and to reach ubiquity. The last section presents the conclusion and future work.

II. Literature Review

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 [6]. Safran, Helic, and Gütl [7] 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 [4]. They are applications with tools to support teaching and learning activities though 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 resources for user communication, such as instant messenger and video-conferences, are becoming common among these environments.

HyperText Markup Language (HTML) is used for any Web application to describe the page interface and its content. Usually, in Web applications where users post text, there is a rich text editor to allow users without HTML skills to write formatted text. In desktop computers, the users use the keyboard to typewrite the letters, and use the mouse to point and trigger text 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 has some improvement defined in the last version, the HTML5, related to support of multimedia, making it easily readable by humans and consistently understood by computers and devices [8]. 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, an 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 Interfaces (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 [9]. The World Wide Web Consortium (W3C) 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 Document Object Model (DOM) 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 of devices equipped with a stylus, such as a tablet, and defines event types for: (i) when a user touches the surface (touchstart), (ii) when a user removes 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 hardware gives flexibility for the developers to define the actions to be triggered for each input data.

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) [10]. The InkML provides a common format for exchanging 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 representing ink gathered by an electronic pen or stylus. It is possible to find some uses of InkML, such as Microsoft Word 2010 which supports electronic ink in text review and the InkML JavaScript Library [11], that offers some functions to allow 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 Web sites use HTML5 to impress users through content exhibition. Few developers consider the user input interaction styles, so they develop Web pages for users with keyboard and mouse on desktop computers which are 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 on 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 on Tablet PCs do not treat electronic ink, so it is necessary to incorporate special applications to treat the electronic ink to have the benefits of the pen interaction style.

Multi-touch in Web applications is more common in games. Johnson [12] presents a tutorial to include features of multi-touch in Web applications, such as handling the touchstart, touchmove and touchend event types. Since we wanted the users to draw with their fingers in touchscreen devices, these event types call functions to start a line, to compose the line, and to stop to draw a line, respectively. To allow multi-touch, it was necessary to store the data from each finger in an array. The browser sends 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 to put the data in the correct line. To avoid the browser from scrolling the page when the user moves the fingers on the screen, the event´s functions *preventManipulation()* and *preventDefault()* were called.

These technologies allow Web applications be adapted considering the device (and considering their input hardware). In particular, mobile devices and wireless networks allow users to interact with a Web application anytime and anywhere.

Modality is a used term to define a mode in which the user data input or a system output is expressed. The communication mode refers to the communication model used by two different entities to interact [13]. Nigay and Coutaz [14] define modality as an interaction method that an agent can use to reach a goal, and it can be described in general terms such as “speech” or in specific terms such as “using microphones”. Bernsen [15] claims that two modalities are not equivalent because they differ in relation to strengths and weaknesses of expressiveness and also in relation to the perceptual, cognitive and emotional systems of the human being. It is also important to understand that device switching can result in changing platform and/or interaction modality. In terms of the system´s usability, therefore, we can find two types of interaction problems when we change devices: those coming from the modality changing and those from the platform changing [5].

For monomodal systems, designers are not limited to choose only one modality. But, in multimodal systems, they can choose many modalities, that, used together, increase the system flexibility and provide other benefits. Interfaces with this characteristic are called multimodal interfaces and the systems are called multimodal interaction systems.

Multimodal systems are present in the Human-Computer Interaction (HCI) literature, and to make them easier to understand and build multimodal systems, some works present a general architecture model for these systems. Multimodality can increase the usability, accessibility, convenience, and flexibility of an application [13], four desirable requirements for e-learning environments. But to build a multimodal e-learning environment it is not a trivial task, it is necessary to deeply understand the e-learning environments, their use and technology, and define an architecture that considers components of multimodal interaction and of e-learning environments, and the Web platform restrictions. To define these components and their communications, we propose an architecture for multimodal e-learning systems.

Some models of Tablet PCs are equipped with touchscreen too, so the user can interact with the keyboard, mouse, track pad, pen or using his/her fingers. Since the touch has become a common way to interact with digital applications, mainly on mobile devices, e-learning environments need to be improved to manipulate data from this input device.

Multimodal interaction is a research proposal to turn the interaction between humans and machines more natural, i.e., closer to the interactions between two humans, and have the benefits to increase the usability, flexibility and convenience [16][17]. Mayes [18] defines multimodal interaction systems as systems with the capacity to communicate with the user by different communication modes, using more than one modality, and automatically gives or extracts meaning. According to Oviatt [13] “Multimodal interfaces process two or more combined user input modes (such as speech, pen, touch, manual gesture, gaze, and head and body movements) in a coordinated manner with multimedia system output”. Lalanne et al. [17] describe multimodal interaction systems, or multimodal systems, that allow users to interact with computers though many data input modalities (e.g., speech, gesture, eye gaze) and output channels (e.g., text, graphics, sound, avatars, voice synthesis). Multimodal systems need to process all input done by the user to identify and process the desired action and generate the output using the appropriate modes. Dumas et al. [16] present a generic architecture for multimodal systems composed by the following components: i) input recognizers & processors, ii) output synthesizers, iii) fusion engine, iv) fission module, v) dialog management and vi) context, user model and history manager. The last four components (components iii, iv, v, vi) make up the Integration Committee.

To implement a multimodal system for Web it is necessary to consider both the multimodal architecture and the Web architecture. Gruenstein et al. [19] present a framework to develop multimodal interfaces for Web, namely, the Web-Accessible Multimodal Interface (WAMI) Toolkit. The framework defines tree client-side components (Core GUI, GUI Controller and Audio Controller) and four more server-side components (Web Server, Speech Recognizer, Speech Synthesizer, and Logger). The user interacts with the Core GUI, described using HTML and JavaScript, and the Audio Controller, a Java Applet to receive the audio input. The collected data is sent to server to be treated by the Speech Recognizer and the Web Server components. The components Core GUI, GUI Controller, Audio Controller and Speech Recognizer can be classified as the input recognizers & processors of the Dumas et al.´s architecture. The Speech Synthesizer can be classified as output synthesizers of the Dumas et al.´s architecture. The WAMI toolkit is focused on speech plus keyboard and mouse modes, but the framework can be expanded to include other modes through definition of new components.

III. PREVIOUS WORK

In our previous work, we developed several tools for TelEduc [3] and Ae [4] e-learning environments. Due the penetration of smartphones in the society, we started to develop tools for these environment that take advantage of the smartphones input hardware and their mobility. The first one was InkBlog, the second was InkAnnotation and the MultiModal Editor was the last one. All tools are described in this section.

A. InkBlog

In e-Learning environments, Weblog [20] is 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.

 ISBN: 978-1-61208-598-2

The InkBlog [21] (Fig. 1) was created to make it easier to handwrite posts and comments in a blog using a stylus in pen-based devices. The approach to develop the InkBlog was to extend the Weblog tool with components to generate and manipulate the electronic ink in the user interface, representing the electronic ink in InkML format. Before that, 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. 1) were done to support input data from stylus. In the architecture, we added a component to receive data from the pen, the InkController component, and a component to renderer this data as electronic ink, the InkRenderer component. Both components, the InkController and the InkRenderer, make up the InkEditor, which is a handwritten text editor for Web pages that renders the electronic ink and receives the input data generated by the 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 by selecting the button options on the right-hand 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 on 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 found inside 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) by 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 hide 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 as she wants and all of them will be stored and will compose the InkML data. When the user finishes to handwrite the post, she will click the "Confirm" button and the generated InkML data will be sent to the server to be stored.

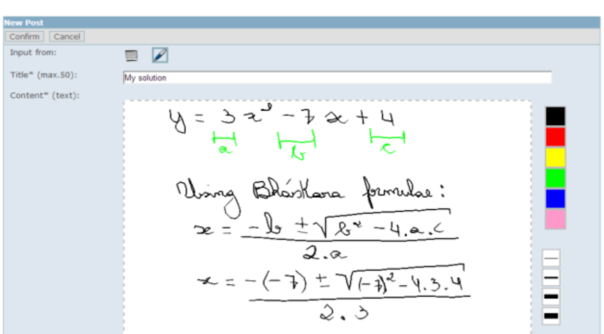

Figure 1. Using InkBlog to handwrite a post using a stylus.

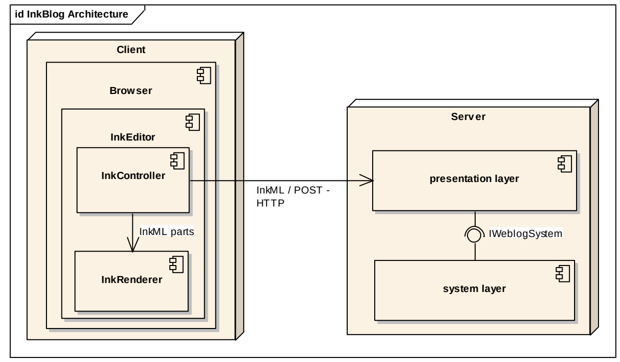

Figure 2. InkBlog components to treat input data from pen [21].

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 on the screen.

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

B. *InkAnnotation*

InkAnnotation [22] is a tool for review of documents, pictures and sketches by handwriting comments using a pen-based tablet or computer (Fig. 3). There are two ways to use this tool. In this first case, the InkAnnotation will be similar to a whiteboard where the user can handwrite or sketch on a blank space or over an uploaded document.

Another use is embedding the InkAnnotation inside

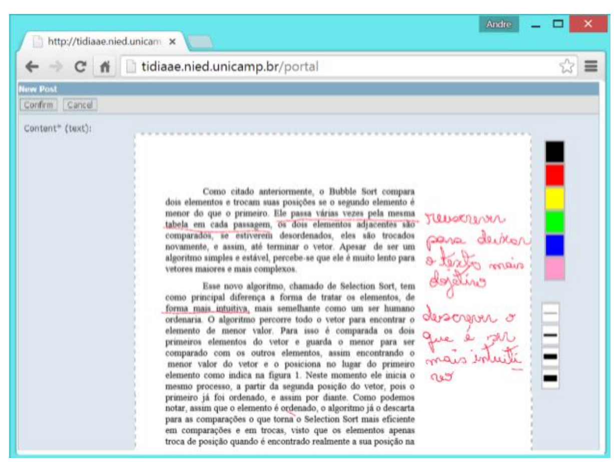

Figure 3. An example of using the InkAnnotation tool to review a document.

another e- tool, e.g., the Portfolio tool. Portfolio is a space each user can use to typewrite an item or do files upload, e.g., PDF files, Word files, and pictures. When the user wants to handwrite a Portfolio item to review it, the user triggers the option "Do Annotation with Ink", and a new window will be open with the document as background. This document will be drawn using the canvas tag, allowing the user to handwrite or sketch over it (Fig. 3).

To treat the data generated by a pen, we reused the InkRenderer and InkController. When the user touches the interface 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 as she wants and all them will be stored and will compose the InkML data. When finished handwriting the review, the user will click in the "Confirm" button and the generated InkML data will be sent to the server to be stored. Since we used HTML5, any browser that supports it can render the electronic ink drawn by the InkRenderer.

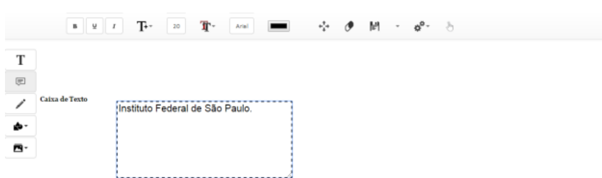

Figure 4. An example of using the Multimodal Editor tool to write a multimodal text.

C. *Multimodal Editor*

The Multimodal Editor (Fig. 4) is a tool for producing multisemiotic texts: texts composed of different forms of representation - images, audios and videos, besides written and spoken language [23][24]. The most common (and old) multisemiotic texts are those that add written text and images and are still widely used today in newspapers, magazines, advertisements. The point, therefore, is that we read more multisemiotic texts than we actually produce such texts. In the development of Multimodal Editor, we assumed that mobile learning is related more to the learner than to technology, since it is the learner who moves. He is the center of learning and the technology allows him to learn in any context [25].

To allow the Multimodal Editor to treat the data generated by a pen, we developed two new components to capture and treat the ink, which are similar to the previous InkRenderer and InkController. However, in this case, the controller generates a SVG draw instead of a InkML file. So, when the user touches the interface within the pen and draws a trace, the controller 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 controller will generate the SVG´s trace node for the new trace. Since we used HTML5, any browser that supports it can render SVG files, so the InkRenderer is the browser´s SVG renderer.

IV. Architecture for Multimodal tools

By analyzing the architecture of each developed tool, we can generalize an architecture for multimodal tools for e-learning environments (Fig. 5). For each modality, a component to receive the data and represent in a way that can be processed and also trigger a system function. In Fig 5.

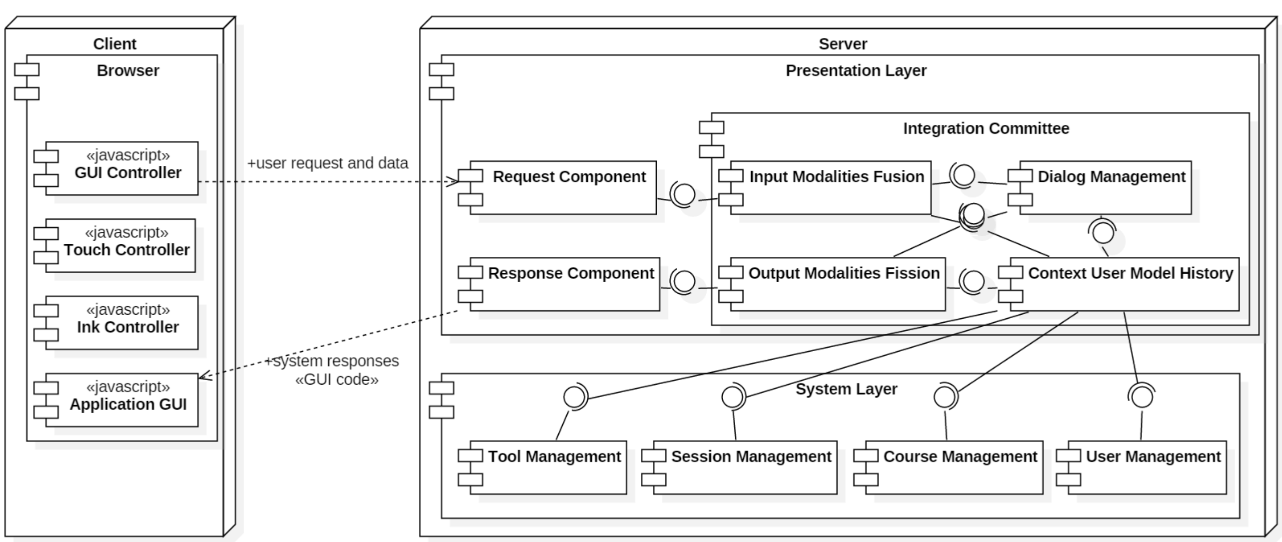

Figure 5. A generic architecture for multimodal tools to e-learning environments.

we specified some components to treat data from touch (Touch Controller), mouse+keyboard (GUI Controller) and pen (InkController), and to render the user interface (Application Renderer) and to render the digital ink (Ink Renderer). The components need to send their data to the server, who receives it by the Request Component (following a Web architecture).

When the devices allow the user to interact with more than one modality, the system can decide how the user can trigger a specific function. This responsibility is that of the Context-aware component and it needs to be configured by the developer since each tool has different functions. The idea is to combine the power of input modalities, e.g., in Tablet PCs. For the 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.

V. CONCLUSION

e-Learning environments are applications that use the Web infrastructure to support teaching and learning activities. To post text, there is a rich text editor to allow users who do not have HTML programming skills to write formatted text. This solution has good usability on desktop computers, but when the user interacts with a pen or by touch, he/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 developed three multimodal tools (InkBlog, InkAnnotation and Multimodal Editor). Based on the knowledge acquired developing these tools, we propose a generic architecture for multimodal tools for e-learning environments, made up of components that treat the data of each modality and perform a system functionality depending on the used modality and available modalities.

As future work, we are developing a context-sensitive component. Since users consider the device´s characteristics, in particular the input hardware, e.g., in devices with pen and touch sensitive screens, users can use a pen to trigger some functions and they can use touch to trigger other functions. In case of devices with only touch sensitive screens, users interact with the fingers to trigger all functions. Another future work aims to provide the developed components so developers of e-learning environment´s tools can improve their tools with multimodality.

ACKNOWLEDGMENT

Authors thank CNPq for grants in the project No. 462478/2014-9.

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 - Distance Learning,” available at <http://www.teleduc.org.br>. [retrieved: Jul. 2013]

[4] Ae Project. “Ae - Electronic Learning 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), IARIA, February 2013, pp. 243-249.

[6] T. O'Reilly, “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]

[7] C. Safran, D. Helic, and C. Gütl, “E-Learning practices and Web 2.0,” Proc. of International Conference on Interactive Collaborative Learning (ICL 2007), Kassel University Press, Sep. 2007, pp. 1-8.

[8] 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]

[9] D. Schepers, S. Moon, M. Brubeck, and A. Barstow, “Touch Events” available at < http://www.w3.org/TR/touch-events/>. [retrieved: Nov. 2013]

[10] Y. Chee et al., “Ink Markup Language (InkML) W3C Recommendation,” available at <http://www.w3.org/TR/InkML/> [retrieved: Nov. 2013].

[11] T. Underhill, “InkML JavaScript Library,” available at <http://inkml.codeplex.com/>. [retrieved: Nov. 2013]

[12] 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-mu lti-touch-and-mouse-input-in-all-browsers.aspx> [retrieved: Nov. 2013].

[13] S. L. Oviatt, “Advances in Robust Multimodal Interface Design”, in IEEE Computer Graphics and Applications, vol. 23, no. 5, Sep. 2003, pp. 62-68, doi: 10.1109/MCG.2003.1231179.

[14] L. Nigay and J. Coutaz, “A Generic Platform for Addressing the Multimodal Challenge”. Proc. of 13th Conference on Human Factors in Computing Systems (SIGCHI 1995), ACM Press / Addison-Wesley Publishing Co., May 1995, pp. 98-105, doi: 10.1145/223904.223917.

[15] N. O. Bernsen, “Multimodality Theory”, in D. Tzovaras (Ed.) Multimodal User Interfaces: From signal to interaction, Berlim, Alemanha: Springer-Verlag Berlin Heidelberg, 2008, pp. 5-28.

[16] B. Dumas, D. Lalanne, and S. Oviatt, “Multimodal Interfaces: A Survey of Principles, Models and Frameworks”, in Human-Machine Interaction, D. Lalanne and J. Kohlas, Eds. Berlin: Springer Berlin / Heidelberg, 2009, pp. 3-26, doi: 10.1007/978-3-642-00437-7_1.

[17] D. Lalanne, B. Dumas, and S. Oviatt, “Fusion Engine for Multimodal Input: A Survey”, in Proceedings of the 11th International Conference on Multimodal Interfaces (ICMI-MLMI´09), ACM Press, 2009, pp. 153-160, doi: 10.1145/1647314.1647343.

[18] T. Mayes, “The ‘M’ Word: Multimedia interfaces and their role in interactive learning systems”, in Multimedia Interface Design in Education, A. D. N. Edwards and S. Holland, Eds. Berlin: Springer-Verlag, 1992, pp. 1-22, doi: 10.1007/978-3642-58126-7_1.

[19] A. Gruenstein, I. McGraw, and I. Badr, “The WAMI Toolkit for Developing, Deploying, and Evaluating Web-Accessible Multimodal Interfaces”. Proc. of 10th International Conference on Multimodal Interfaces (ICMI 2008), ACM Press, Oct. 2008, pp. 141-148, doi: 10.1145/1452392.1452420.

[20] J. Ray. “Welcome to the blogosphere: The educational use of blogs”. Kappa delta pi Record, 2006, pp. 175-177.

[21] A. C. da Silva, 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.

[22] A. C. da Silva, “InkAnnotation: An Annotation Tool for E-Learning Environments,” Proc. of The 2015 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE 2015), Universal Conference Management Systems & Support, July 2015, pp. 73-74.

[23] F. M. P. Freire, F. L. Arantes, A. C. da Silva, and L. E. L. Vascon, “Estudo de viabilidade de um Editor Multimodal: o que pensam os alunos?,” Proc. of the XX Congreso Internacional de Informática Educativa (TISE 2015), v. 11, Universidad de Chile, Dec. 2015, pp. 109-119.

[24] F. L. Arantes, F. M. P. Freire, Jan Breuer, A. C. da Silva, R. C. A. de Oliveira, and L. E. L. Vascon, “Towards a Multisemiotic and Multimodal Editor”, Journal of Computer Science & Technology, v. 17, n. 2, oct 2017, pp. 100-109.

[25] M. Ally and J. Prieto-Blázquez, “What is the future of mobile learning in education? Mobile Learning Applications in Higher Education”, Revista de Universidad y Sociedad del Conocimiento (RUSC), v. 11, n. 1, 2014, pp. 142-151. DOI: http://dx.doi.org/10.7238/rusc.v11i1.2033.