

- ORIGINAL ARTICLE -

Towards a Multisemiotic and Multimodal Editor

Flávia Linhalis Arantes¹, Fernanda Maria Pereira Freire¹, Jan Breuer², André Constantino da Silva^{1,3}, Renato César Alves de Oliveira¹, and Lucas Eduardo de Lima Vascon¹

¹*Nucleus of Informatics Applied to Education (NIED), UNICAMP, Campinas, Brazil*
{farantes, ffreire}@unicamp.br, renato.oliveira2808@gmail.com, lucasvascon@outlook.com

²*University of Siegen, Nordrhein-Westfalen, Germany*
jan.breuer@uni-siegen.de

³*Federal Institute of São Paulo, Hortolândia, Brazil*
andre.constantino@ifsp.edu.br

Abstract

Mobile devices, such as tablets and smartphones, are increasingly present in the everyday life of young people. Despite such insertion, the use of those devices in scenarios such as m-learning is still a missed opportunity. In a previous work we verified that an editor that aggregates various kinds of media can help students in their studies. With that motivation in mind, we realized the Multimodal Editor – a useful tool to leverage the multimedia authorship. In this paper we describe our trajectory toward the Editor – requirements gathering, implementation, challenges, usage scenario and evaluation.

Keywords: Mobile devices, Multimodality, Multisemiotic, Software requirements.

1 Introduction

Mobile devices became increasingly popular; most of them have touch screen displays, are easy to carry, have autonomy for hours, and enough computing power to process web pages, audio and video files. Due to the growing use of smartphones and applications for these devices, it is expected that students use these resources in their learning activities, increasing the number of people who benefit from mobile learning (m-learning).

However, m-learning is not a reality in many countries yet. Some institutions have mobilized in order to change this scenario. In 2014, UNESCO launched a set of guidelines with the objective of assisting the policy makers to better understand what is mobile learning and how its benefits can be used “as leverage

to advance progress toward education for all” [1](p. 7). The guidelines have wide application and can adjust to a wide range of institutions, including pre-schools, schools and universities, community centers, technical and vocational schools, and were organized with experts from 20 countries. According to the data presented in the document, in 2013 the number of mobile devices with Internet – most of them cell phones – surpassed the world’s population. Still, despite its undeniable presence and the different types of learning that the m-learning can support, it is usual that mobile technologies, especially smartphones, are forbidden or ignored in the formal systems of education, which “represents a missed opportunity” [1](p. 42). According to UNESCO, the mobile devices are not a new passenger and as the power and functionality of mobile technologies continue to grow, its usefulness as educational tools will probably increase and, it is going to have a central role for education [1](p. 42).

To get an idea of the number of young users of mobile devices, we present some data published in 2014 by the Telefônica company, responsible for mapping 6,700 behaviour of young people around the world in relation to political, economic and innovative positions, including the use of mobile technologies [2]. Taking Brazil as an example, 78% use smartphones and 42% use tablets. These rates are comparable to those of Western Europe and the United States, surpassing the use of desktops (69%). 68% of Brazilians use smartphones to access social networks, but only 5% use smartphones for m-learning activities – distance courses, discussion forums, etc. It is interesting to note that a considerable portion of young Brazilians (37%) uses the smartphone to create some type of content, mostly visual (pictures and videos).

We realized that a Multimodal Editor could be a useful tool to leverage this type of authorship. More than that, our goal is to promote, in a simple way, the “multimedia authorship” [3]. We assume it can be supported and amplified by the Multimodal Editor, a free web application, given its multisemiotic potential for multi device, cross-platform, and multimodality.

Citation: F. Linhalis Arantes, F. M. Pereira Freire, J. Breuer, A. C. da Silva, R. C. A. de Oliveira, L. E. de Lima Vascon. “Towards a Multisemiotic and Multimodal Editor”. *Journal of Computer Science & Technology*, vol. 17, no. 2, pp. 100–109, 2017.

Received: February 2, 2017 **Revised:** July 04, 2017. **Accepted:** July 11, 2017.

Copyright: This article is distributed under the terms of the Creative Commons License CC-BY-NC.

We define our Multimodal Editor as a space to write, to draw, to insert images, to add audio/video and to insert links as a way to record and produce information using different types of media. With such a simple definition people may think that it's just another multimedia editor. However, the Multimodal Editor that we describe in this article goes beyond. It can be defined based on five features that, together in one product, make it different from other editors found in the literature and on the market. These features are the following:

- **Open source:** different from most of the available multimedia editors, our editor is free software and, as such, we intend to have a community of developers and users.
- **Multi device:** The editor should allow capturing a variety of media using desktops or mobile devices.
- **Cross-platform:** The editor is created as a web application to realize a platform independence, which enables a user to use all kinds of available hardware.
- **Multimodal:** The editor should aggregate various interaction modes – touch, pen, keyboard and mouse inputs.
- **Multisemiotic (or multimedia):** The goal of the multimodal editor is to support the development of multimedia documents, compatible with contemporary practices. In this article, we prefer to use the term multisemiotic instead of multimedia because we're more interested in how people produce sense through this type of document than in simply media usage.

The Multimodal Editor can be used as a web application in general, in scenarios where there is a necessity to register information with mobile devices using various types of media, such as classroom or in business meetings.

In a previous work, we chose an educational scenario to carry out a feasibility assessment. Our objective was to know if an editor like that is something desired by students, assuming that it can be helpful to take notes in class and/or to revise content later. The results were very positive for the following reasons [4]:

- We discovered that the most common form used by students to register content in the classroom is to take notes in notebooks. Among those who use smartphones we found students who often record audio class for later transcription and/or photograph drawings and schemes made by the professor in the whiteboard. Among those using tablets, in addition to audio and image storage,

they also take notes using stylus or keypad. Students are used to study using different semiosis, i.e., different forms of content representation, a process that can be called multisemiotic [5, 6]. This leads us to conclude that an editor that aggregates various kinds of representation, i.e., various types of media can help students in their studies.

- Most students already use a mobile device to accompany the classes, often in conjunction with a notebook.
- Students consider that it is useful to have a multisemiotic document for later study.
- Most students think that the use of mobile devices facilitates to accompany classes (53.8%).
- In order to register information during class, many students switch between applications. An editor that can integrate various types of media was pointed out as something useful and desired.

2 Related Work

Before starting our Editor's implementation, we analyzed some applications for mobile devices with the potential to produce multisemiotic documents. The purpose of such an investigation was to know better the similar editors, as well as to check interesting usability features that could be integrated in our editor.

All the tests were conducted by the same user with the Samsung Galaxy Tablet Note Model GT-8000 10.1, with Android version 4.4.2. We verified if the multimodal editors available on the market support or not, in a single document, the following features:

- To insert written text and/or images (through physical or virtual keyboard),
- To take photograph and/or to insert pictures,
- To attach and/or record audio,
- To attach and/or record videos,
- To insert links,
- To draw freely (including handwriting),

The results are shown in Table 1. The column Functionalities and Medias shows the analyzed functionalities and the column named Operating Systems shows the compatibility between the application and the OS families (iOS, Android, Windows) and if it works on the web.

Android is the platform that supports the wider number of applications. The applications indicated on columns Operating Systems work on the following devices: iPad and iPhone (iOS); Smartphone and tablet (Android), Windows Phone and Windows OS (Windows). Evernote is the only application that provides a

Table 1: Operating Systems, Functionalities and Medias available in mobile devices apps.

Apps	Operating Systems			Functionalities and Medias									
	iOS	Android	Windows	Web	Text	Image	Take picture	Attach audio	Record audio	Attach video	Record video	Link	Free drawing
FiiNote		X			X	X	X	X	X	X	X	X	X
Evernote	X	X	X	X	X	X	X	X	X	X		X	X
Google Keep		X		X	X	X	X	X	X			X	
OneNote	X	X	X	X	X	X	X		X			X	X
Skitch	X	X	X		X	X	X						
MyScript Smart Note	X	X			X	X	X						X
Squid		X	X		X	X	X						X
Genial Writing2	X	X			X	X	X						
Write		X	X			X	X						X
Notepad+		X			X	X							X
Lecture Notes		X			X								X

browser extension. Evernote, Google Keep and One Note can be accessed through web sites.

We can see on Table 1 that the first two applications, Evernote and FiiNote, are the most complete in terms of semioses: they support text, image, taking pictures, attaching and recording audio, inserting links and drawing freely. The difference is that the first one (FiiNote) allows to attach videos and to record them directly (with Evernote you cannot record video). Google Keep does not support video and free drawing; while One Note does not support attaching audio and attaching or recording video. Skitch, MyScript Smart Note, Squid and Genial Writing 2, allow adding text, image, photo and free drawing. Write, Notepad + and Lecture Notes are more restricted applications.

Our survey also included which interaction modes are supported by applications. It is important to remember that the tests were executed with Samsung Galaxy Tablet Note, which offers touch, pen and virtual keyboard. All applications reviewed support touch interaction; all support keyboard interaction, except MyScript Smart Note and Write; with respect to interaction with pen, just Google Keep, Skitch, Genial Writing2 and Notepad + do not support.

Our expectation is that Multimodal Editor can register and produce information using different types of media. So, the editor should support various modes of interaction (touch, pen, keyboard, and mouse) and work across multiple platforms (operating systems and browsers accessed by laptops, tablets and smartphones). We also intend that the editor can be available in the web and have open-source code, important differentials when compared to other editors presented in this section. These last two features provide flexibility of use to the editor, because it can be improved and reused by web applications in general.

In addition to this study, we made a preliminary investigation about open-source editors, which could be reused to the development of our editor. The most relevant ones are the following:

- SVG-edit: allows to create geometric shapes or use ready-made shapes, draw freely, import images and insert gridlines. It is the most similar open-source project we found because it works with the SVG standard (see section 5).
- Etherpad: is a collaborative text editor. The collaboratively functionality is something desirable in future versions of our editor.
- Etherdraw: is a collaborative drawing editor with a minimalist interface, suitable for laptops, tablets, and mobile phones. It has many features in common with our editor, but it uses canvas while we use SVG.
- Firepad: is an open source collaborative code editing. It can be embedded into applications by adding a snippet of HTML code. This is an interesting feature we want to develop in future versions of our editor.

3 Series of activities for requirements gathering

We planned and executed a series of activities to help gathering requirements to Multimodal Editor. The activities were held on May 22th, 2015. Seventeen people participated with the following roles:

- 1 teacher who presented a 15-minute lecture, using multimedia projector and whiteboard.

- 13 students who made class notes, each carrying a mobile device that could be a smartphone, a tablet or a laptop.
- 3 researchers who conducted the series of activities.

The series of activities was divided in two stages: in the first one the professor simulated a lecture with an average time of 15 minutes, using slides projection and whiteboard. The slides were made available to the students who wished to accompany the lecture and take notes on them. Each one of the 13 students had a mobile device that should be used to register the class and to make a useful material for later studies – the students could take notes, record audio end/or video, take pictures and make draws. Four students used laptops (keyboard and mouse interaction), 5 of them used tablets (pen and touch interaction) and 4 used smartphones (interaction with touch). The objective of this first stage, therefore, was to collect data about how students gather information during class, how they use the devices to register the class, and how they access the stored material for further study.

In the second stage the students were divided in four groups in order to develop a multisemiotic document in Power Point using the information captured in stage 1. The groups were formed with students who had used different devices in stage 1. The objective of stage 2 was to collect data about how students accessed the notes previously stored and how they used that notes to create a multisemiotic document.

Three methods were used to collect data: (i) recording of observations made by researchers along the activities, (ii) analysis of questionnaires applied at specific times of the activities and (iii) analysis of films. In addition, at the end of the activities, we proposed a quick brainstorming, in which participants were asked to say what they found “easier” and “harder” in relation to the proposed activities. More details about the series of activities can be found at Freire et al. [4].

4 Editor’s Requirements

In this section, we show how the results of the series of activities were used to gather the editor’s requirements.

During the series of activities, the students were purposefully without notebooks because we wanted to see how they would register the class with mobile devices. We collected information about the devices’ functionalities used to accompany the class. Our observations show that all students made text edition – only one of them did not use a text editor, but he used the edit function of the Power Point. A reasonable portion of the students (30.8%) used a free drawing editor and/or took pictures.

As we expected, the preferable way for students to document the class is to write, either by hand or

through the keyboard. This result was already expected, because writing is the traditional mode already practiced with or without the use of technology. With those results in mind, we concluded that the editor must offer features for text typing and pen-based writing.

Before lecture, the teacher made the slides in .ppt and .pdf formats available. Most of the students accessed the material to accompany the lecture (61.5%), but only 15.4% made notes on them using the mobile device. Most students made notes in different applications and referenced the material. This means that the participants alternated their attention between at least two applications: the one that displayed the slides and other one used to take notes and to register other types of media (images and links, for example). This applications alternation necessity must be considered in the design of Multimodal Editor:

- The system has to allow images insertion and make notes on them (the professor’s slides, for example).
- The system has to allow links insertion.
- The system has to allow the insertion of different types of media, sequentially, on the same line.

During the activities, we had questions about how students used to register information in the classroom, in order to facilitate their study later. Most students write (92.3%) and/or pay attention to the teacher’s explanation (84.6%). About 15.4% recorded the class, 7.7% took pictures and 15.4% made notes on the slides during class. We asked the participants if the use mobile devices facilitated the accompanying of the class. The results were as follows:

- 23% asked no. They think that the best method to capture information in a class is to use the notebook. They also think that mobile devices can disperse the students, mainly the smartphone.
- 23% asked maybe, because it depends on the suitability of the device and the application.
- 53.8% think yes, due to reasons such as: agility in gathering information, mobility, greater variety of information collected and the possibility of doing research in other sources on the topic of the lesson.

The ones who responded “no” or “maybe” (46.2%) think that the device could harm the student’s attention with respect to the content that is being taught in class. So, we can say that the handling of parallel equipment during the class is the big factor of distraction. Therefore, we need to invest in simplicity and usability to facilitate the task of the user. With that in mind, we draw the following requirements:

- The system has to allow the user to take a picture and insert it instantly in the editor.
- The system has to allow the user to insert audio and video directly in the document. Thus, the user does not need to search for an audio recorder application or a video camera.
- The system has to allow the user to undo and redo actions – this allows easy and fast correction of errors.
- The editor has to be simple and intuitive for rapid learning of its main functions, avoiding distraction.

We asked about the contribution of the mobile device to the study process. Seven answers (63.63%) were related to the possibility of having differentiated information, the possibility of bringing together in a single file various information, to draw, to capture the information in different formats, to write and to capture images, to realize parallel activities faster (making notes, drawing, searching for information, etc.).

Although 46.2% of participants have asserted that they distracted in class due to the use of the device, when we asked if the equipment has helped in the study, 84.61% said yes, which corresponds to 11 of the 13 volunteers. One of the participants who claimed that the mobile device did not help said, however, that the produced document will help later, at the time of reviewing content.

As we can see, virtually all students want to use the multimodal document produced with the multimodal editor to study later. Therefore, the following requirements are important:

- The system must provide an option to save a document.
- The system must provide a way to save a document automatically on the server. The system itself will make backups for that, if there is a failure, the user does not lose everything that has been done.
- The system should provide options to download a document in PDF format.

After using the series of activities results to gather the editor's requirements we made the documents of requirements [7]. There are 46 functional requirements related to the following features: to save documents, to download documents, to insert images and to take pictures, as well as to perform operations on images, to make free drawing and to perform operations on it; to insert text, link, audio, video, and to perform operations on all these elements. In addition, there are requirements to insert grid lines, redo, undo, cut and paste operations.

5 Design and implementation issues

Considering the variety of mobile devices such as desktop-PC, laptop, tablet, smartphone, iPhone, windows phone, etc., it is not possible to write one application that works on all devices perfectly. In order to offer a wide area of supported devices, it was decided to program a web application that enables the user to interact and to use the Multimodal Editor with nearly every device as long as it has connection to the Internet and works properly with HTML5 and CSS3, the currently used programming languages for web applications.

The Multimodal Editor was created as a web application to realize a platform independence, which enables a user to use all kinds of available hardware. Therefore the editor is based on a responsive web design. Responsive web design is a concept initially proposed by Marcotte [8] which aims to design and implement web user interfaces in order to have a great viewing experience in the device that will render the page.

As a web application a high diversity exists by the usage of the Internet browser. One of the most complicated challenges is to support the different kinds of browser implementations used by the amount of supported devices. Therefore a limitation of supported browsers is necessary (IE, Firefox, Chrome, Safari, Opera). The most serious issue to deal with are the individual listener concepts by each browser.

Touch screen devices, in particular, have their own commands forcing us to implement the Multimodal Editor's functions for each necessary browser itself – especially the touch-interaction. According to the W3C, they are working on a standard for touch-interaction on the web but, until recently, the published standard is still an editor's draft and the browsers are not forced to support the standardization [9]. Nowadays, HTML5 means the standard for user-oriented Internet and has become a brand with its own logo from the W3C [10](p. 38), but a standard for interaction with touch-screens does not exist yet. In order to implement the Editor, HTML5, CSS3 and JavaScript are used on the client-side and PHP and MySQL on the server-side.

As mentioned in the previous section, the Multimodal Editor has to offer various types of inserting and saving data. HTML5 offers for the first time its own object called canvas, which supports drawing, writing etc. on the browser. The inserted data is created as pixel and is only available as long as it is within the canvas object. Moving an element out of the canvas will delete it if it was not saved before. Therefore it was decided to implement the functions of the Multimodal Editor in SVG (Scalable Vector Graphics). The main point to decide against the Canvas-Object is the required permanent storage and permanent manipulation of the inserted texts, drawings and images. The common SVG-Object maintains the necessary manip-

ulations and storage requirements. Out of that the conversion from SVG to PDF, from SVG to XML/JSON, from XML/JSON to SVG is challenging, because no standardization of listener concepts exists.

Another point we had to think about was a design/layout and a way of performance that (i) offered enough space to insert data in the editor, (ii) made it easy to change between different kinds of input (text, graphic, audio,...), (iii) was intuitive to use. Therefore it was decided to implement an editor that included all possibilities to insert semiotic data in one design and accepted the limited surface based on a big menu (see Figure 1).

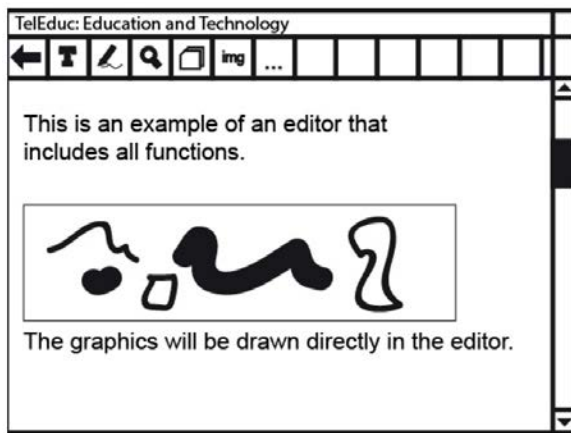


Figure 1: An editor that integrates all semiotic input functions.

A design challenge is to present all necessary functions on a big display as on a laptop the same way as on a small smartphone display. Especially editing the same document easily with different devices and different display resolutions is a huge challenge. Implementing optimal viewports for each display size is important.

The most complicated challenge is to maintain the integrity. Without an existing Internet connection the editor cannot be used and with losing the connection during the editing process the document cannot be stored securely on the database. A possibility to save the document locally and upload the changes with the next existing Internet connection is important.

6 Editor's Implementation

We implemented a prototype version which we called EdiMM – the latest version can be accessed at <http://fenix.nied.unicamp.br/EdiMM/>. Figure 2 shows its current interface (June 2017) in a smartphone screen.

EdiMM has a clean and simple interface. There are two menu bars – on the top and on the left. The top menu bar has options for line thickness selection, color palette, moving and deleting objects, saving files, creating new files and grid lines. In the left menu bar

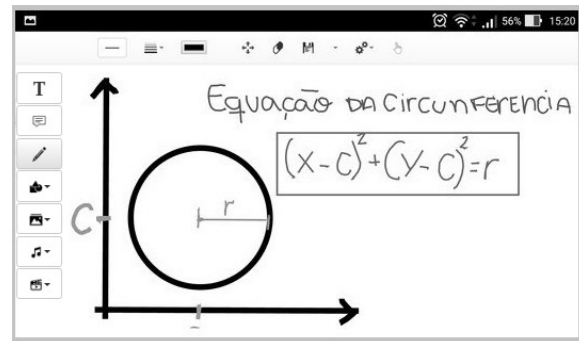


Figure 2: EdiMM multimodal editor interface in Asus zenfone 2 smartphone.

there are options to insert text, to insert free drawing, to insert pre-defined geometric shapes and to insert pictures.

Next, we describe the features that are implemented, partially implemented and not implemented:

- Implemented: save SVG document on the server, insert images in the most usual formats, delete pictures, do free drawing, use predefined shapes for drawing, moving and deleting drawings, choosing colors and thicknesses of lines, enter text, move text, select sizes, types, colors and highlights, insert gridlines, move selected parts, download the document in PDF format.
- Partially implemented: features for working with audio and video, object's selection.
- Not implemented: delete text, document autosave, document download in SVG format, take a picture and insert it into the document, resize drawings, features for working with links, undo and redo operations, cut/copy/paste operations.

7 Usage Scenario and Evaluation

In this section, we present a usage scenario to better illustrate EdiMM's functionalities, as well as to identify limitations and improvements needs. For a better organization, this section is divided into usage scenario, tasks performed and evaluation.

7.1 Usage Scenario

The usage scenario was a meeting of a software development team – the developers should use EdiMM to register the meeting. A project manager and six web developers attended the meeting. The role of each participant was as follows:

- Project manager: responsible for conducting the meeting.
- Experienced users: these participants knew well the features and limitations of EdiMM. One of

them used a smartphone and the other used a laptop to register the meeting.

- Intermediate users: participants that had already tested some features of EdiMM and partially knew about its use. One of them used an iPad and the other used a laptop to make annotations.
- First time users: novice users, who have not had contact with EdiMM before. One of them used an Android tablet and the other used a smartphone.

The room where the meeting took place was equipped with a Smart TV (without datashow) and a white board. The project manager exhibited an image on the TV with a web page first version and wrote on the white board the name of the web components that would be part of that page. The purpose of the meeting was to discuss the componentization of the web application that was displayed on the board – to discuss the functionality of each component and who would be responsible for developing each component. The participants should use EdiMM to register all these information. By the nature of the meeting, EdiMM was expected to be a good tool for the registry because the participants had dispersed information – an image (on TV), notes (in the white board) and an explanation given by the project manager (audio or video). We expected that the participants could have all those multisemiotic information in a single document.

Figure 3(a) shows the web page displayed in the Smart TV in the beginning of the meeting, and Figure 3(b) shows the notes on the white board at the end of the discussion (with information about the functionality of each component and who will be the developer).

7.2 Registering the meeting with EdiMM

Before starting the discussion on the components, the project manager showed EdiMM to the participants with a brief explanation. She suggested them to use the image projected on TV as a background, so the notes could be made on the image. She also suggested the participants to record an audio or video with the explanation of the functionality of the components.

The discussion on the components lasted about 15 minutes. For organization purposes, we divided the information registry using EdiMM in four tasks: insert background image, make notes on the image, insert audio/video and retrieve the document. In the remainder of this section we describe the experience that the participants had with EdiMM during the execution of these tasks.

Task 1: Insert background image

Experienced users had no problems to insert the background image. However, other users were unable to do it readily because it took a while to realize that it was necessary to set a point on the screen where the

image should appear, and then drag the image to resize it. After they found that out, they were able to insert the image. Only one of the users (a beginner) had great difficulty to insert the image, because he could not locate it in the file system of the Android tablet.

Task 2: Make notes on the image

Users who were with laptops were the ones who best managed to make notes, because the use of the keyboard facilitated the task. The other participants used the finger or a pen – one of them said the pen was too thick and there was no way to change that.

Figure 4 shows the notes made by an experienced user with a laptop and Figure 5 shows the notes of an intermediate user with an iPad. We can see that the annotations made with the laptop are far more organized, because they were made with EdiMM's text box and shape tools using keyboard and mouse, while the notes made with the tablet give the impression to "mess up" the document, because they were made with EdiMM's free drawing tool using pen/finger.

Task 3: Insert audio or video

The participants were able to insert video and audio files in EdiMM, however they had problems to (re)position and to retrieve the audio/video.

Task 4: Retrieve the document

At the end of the meeting, the project manager asked the participants to close the document and to (re)open it. As we explained in section 5, EdiMM depends on a good Internet connection to store the document without losing data. As there were several people recording audios and videos at the same time, some users lost data. Figure 6 shows the same image on Figure 4 – when the user recovered the document, the last annotations and the recorded audio were lost.

7.3 Evaluation

At the end of the meeting, the participants responded to a questionnaire to evaluate the experience they had with EdiMM. Only one of the participants did not answer to the questionnaire as it was not possible to insert the background image, so he did not do the notes during the meeting.

Firstly, we asked what the users thought about the interface intuitiveness – the answers could be "Yes", "Yes, but some improvements are needed", "little intuitive", or "No". One of them answered "Yes" (20%), three answered "Yes, but need improvements" (60%) and one replied that the interface was "little intuitive".

About the clarity of icons, four users answered that they are very clear (80%) and one thought they are little clear (20%). About the layout of EdiMM (organization of icons and menus), one user answered is well

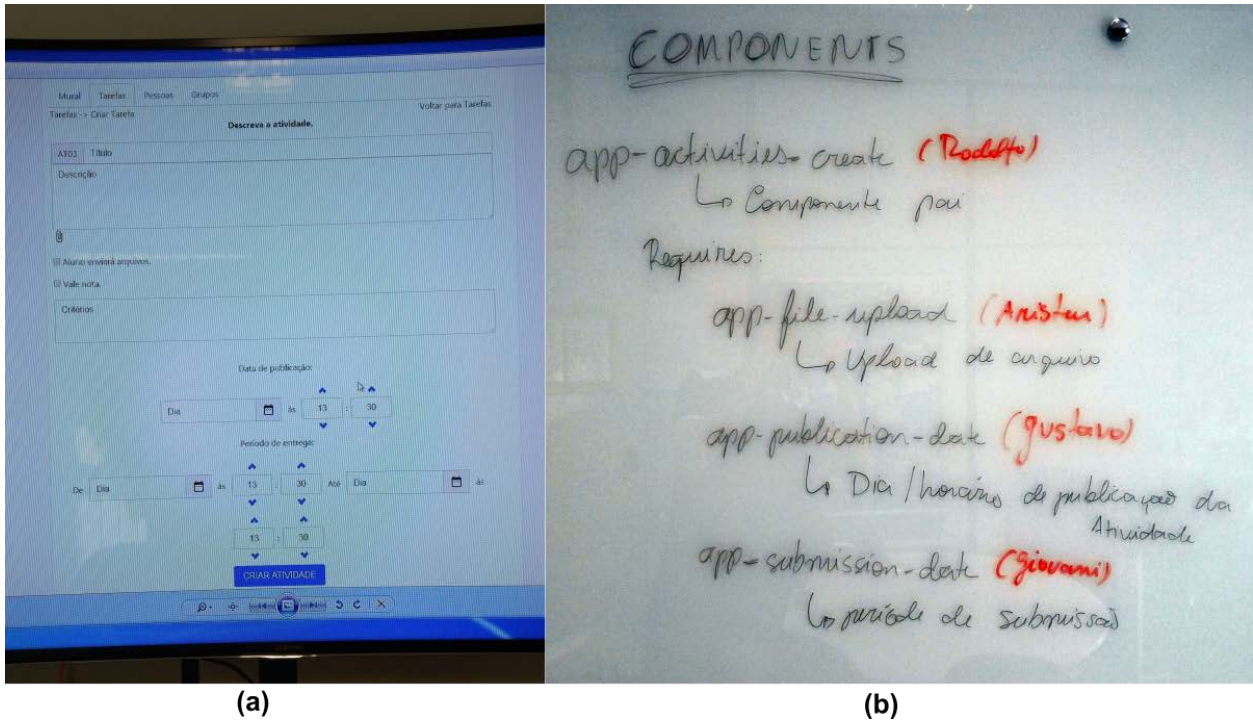


Figure 3: (a) web page displayed in the Smart TV, (b) notes on the white board in the end of the meeting.

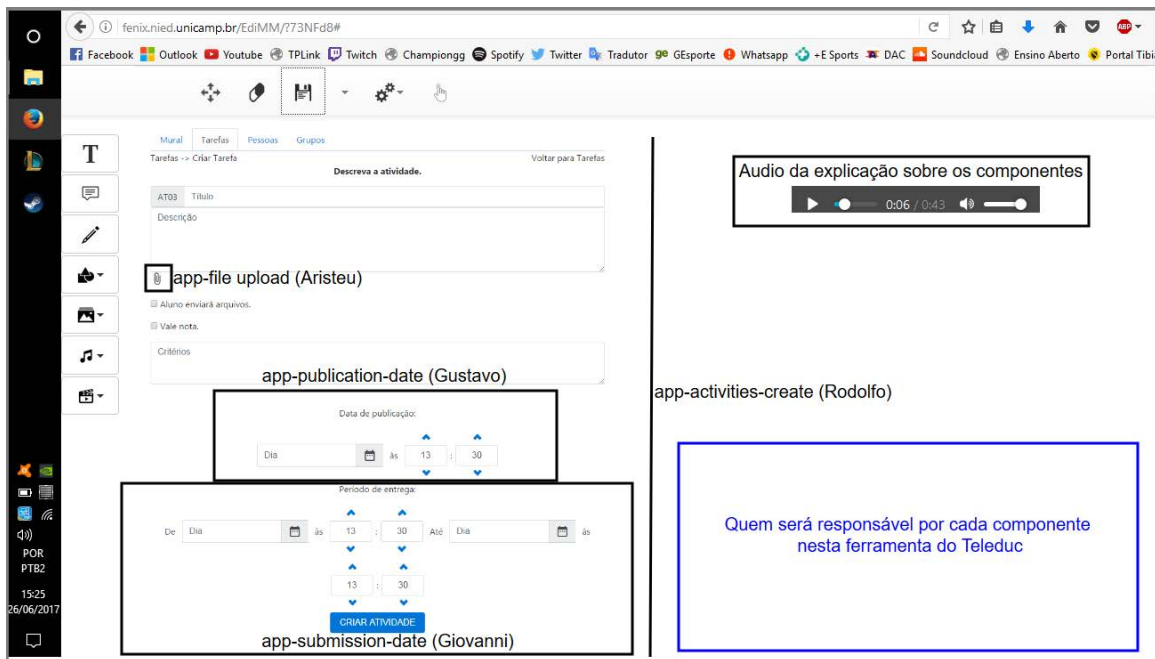


Figure 4: Annotations made by an experienced user with a laptop.

organized (20%), three thought is well organized but need improvements (60%) and one user thought it was not well organized.

We also asked them about the bugs found during the interaction. As this is a prototype version of Ed-IMM, with some features still under development, it was expected that users find bugs or features not yet finalized. The following bugs were the most reported by the users: images insertions generates more than one

image if you click without dragging, audio and video were lost when retrieving the document, the insertion of text generated more than one text box, some objects were behind the image and there was no way to bring them to front, audio and video were not inserted in the tablet, problems to move objects, PDF document was not generated correctly and sometimes the eraser did not work.

We asked what features they used more and if they

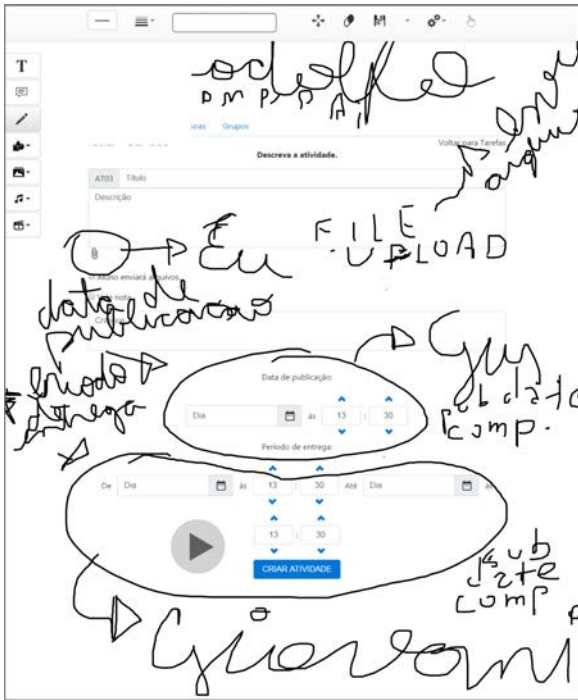


Figure 5: Annotations made by an intermediate user with an iPad.

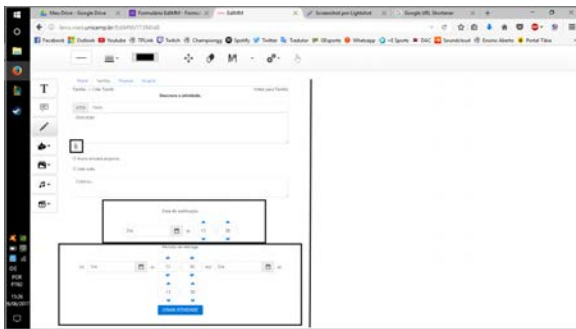


Figure 6: Same annotations showed in Figure 4. The annotations in the right side were lost when the document was retrieved.

had problems in the usage. The most used features by the participants with the smartphones were text box, freehand annotations (pencil tool), images, audio and video insertion. They reported some difficulty to use the text box tool and to take notes by hand, because “*the screen is too small to make this type of annotation*”. The features most used by laptops users were pre-defined shapes, text box, images, audio and video insertion. They had difficulty to move and resize objects. The most used tool by the participant with the tablet was freehand annotations (pencil tool). He had difficulties with the insertion of objects, because several touches were needed to complete the action. He also had difficulties to insert audio, to find the option to change the color of the line, and to move objects.

We also asked if the users missed some function-

ality in EdiMM during interaction and why. One of the smartphone users said he would like more agility, because meeting situations require agility to follow what is being said. The other smartphone user said he has not missed additional features. One of the laptop users said he missed a tool to select elements, regardless of the type – for example, select an audio, a photo, a video, and import all together in the editor; so EdiMM might “be aware” of the type of each file and open the appropriate element to display on the screen. According to him, this functionality could speed up and facilitate the work at a meeting, because the user should only change the position and size of these elements. The tablet user said EdiMM has a selection tool, but it is not clear which object is selected – the user suggested to highlighted the selected objects with a dashed line for example. He also said it was hard to know which tool was selected, he suggested to make a stronger highlight to the tool currently in use. Finally, he said EdiMM should automatically recognize the input device, without the need to click on the multitouch button (currently, it is necessary to click that button whenever the interaction is by touch).

Finally, we asked the opinion of the participants about the usage of EdiMM in this scenario (to capture information from a meeting and to register notes for future reference). Users of smartphones thought that EdiMM is a good tool for this task, as it offers a variety of functionalities to support such a scenario. The laptop users also agree that EdiMM is suitable for the scenario, but it needs improvements because some features are not still working well. The tablet user answered he had many difficulties during the annotations, so he does not agree that EdiMM is a good tool for a while.

8 Conclusion and future work

In this article we described our trajectory toward a Multimodal Editor, which has features that, together in one product, make it different from others encountered in the literature and in the market – open source, multi-device, cross-platform, multimodal and multisemiotic.

We started our trajectory with an investigation about other editors that could be considered multimodal and multisemiotic. Afterward, we conducted a series of activities to gather requirements, which were formally documented. We implemented a prototype version named EdiMM, available at <http://fenix.nied.unicamp.br/EdiMM/>.

EdiMM were tested with a usage scenario of a software development team meeting – the developers used the editor to capture information from the meeting and to register notes for future reference.

As EdiMM has been evaluated in a version that is considered a prototype, users found several bugs and had some difficulty with features not yet finished. Despite of this, the evaluation was positive, because it

shows that most users consider EdiMM a good tool for that kind of scenario, as it offers a variety of functionalities to support the described tasks.

The development of an editor with such features introduces various challenges, such as platform independence, browsers particularities, integrity maintenance, among others. Our next step is to work around these issues and to consider the suggestions and bugs reported by users.

The Multimodal Editor is free software, its source code is available in GitHub at <https://github.com/nied-unicamp/EditorMm>. So, another future direction is to create a community to leverage the participation of volunteers.

Acknowledgements

We thank the users who participated in the tests and evaluation of the Editor.

Competing interests

The authors have declared that no competing interests exist.

References

- [1] UNESCO, “Unesco policy guidelines for mobile learning,” 2013. Available at: <http://unesdoc.unesco.org/images/0021/002196/219641e.pdf>. Retrieved February 2nd 2017.
- [2] Exame, “78% dos jovens usam smartphone, mostra pesquisa,” 2014. Exame.com online magazine. Available at <http://exame.abril.com.br/tecnologia/noticias/78-dos-jovens-usam-smartphone-mostra-pesquisa>. Retrieved February 2nd 2017.
- [3] J. D. Bolter, *Handbook of Literacy and Technology*, ch. Hypertext and the Question of Visual Literary. Lawrence Erlbaum, 1998.
- [4] F. M. P. Freire, F. L. Arantes, A. C. Silva, and L. E. L. Vascon, “Estudo de viabilidade de um editor multimodal: o que pensam os alunos?,” in *In: 20th Congresso Internacional de Informática Educativa (TISE 2015). Proceedings of TISE - Nuevas Ideas en Informática Educativa*, vol. 11, (Santiago, Chile), pp. 109–119, 2015.
- [5] A. B. Kleiman, *Preciso ensinar o letramento? Não basta ensinar a ler e escrever?* Campinas, UNICAMP: Secretaria de Ensino Fundamental, 2005.
- [6] R. Rojo, *Letramentos múltiplos, escola e inclusão social*. São Paulo: Parábola Editorial, 2009.
- [7] R. S. Pressman and B. R. Maxim, *Software Engineering: a Practitioner’s Approach*. New York: McGraw-Hill Education, 8 ed., 2014.
- [8] E. Marcotte, “Responsive web design.” Ebook. A Book Apart (Ed.), 2011. retrieved February 2nd, 2017, from <https://abookapart.com/products/responsive-web-design>.
- [9] D. Schepers, S. Moon, M. Brubeck, and A. Barstow, “Touch events. w3c editor’s draft,” May 2014. Retrieved February 2nd, 2017, from <https://dvcs.w3.org/hg/webevents/raw-file/tip/touchevents.html>.
- [10] F. Franke and J. Ippen, *Apps mit HTML5 und CSS3. Für iPad, iPhone und Android aktualisierte und erw. Aufl.* Bonn: Galileo Press (Galileo computing), 2 ed., 2013.