

Collaborating Using an Interactive Whiteboard: A Case Study on the Learning of Logic Circuits

Eduardo Velloso
Lancaster University
School of Computing and
Communication
South Drive, LA14WA
Lancaster, UK
e.velloso@lancaster.ac.uk

Denise Filippo
Rio de Janeiro State University
Superior School of
Industrial Design
Rua Evaristo da Veiga, 95
Rio de Janeiro, Brazil
dfilippo@esdi.uerj.br

Hugo Fuks
Pontifical Catholic
University of Rio de Janeiro
Department of Informatics
Rua Mq São Vicente, 225
Rio de Janeiro, Brazil
hugo@inf.puc-rio.br

ABSTRACT

Even though interactive whiteboards have been widely used in classrooms around the world, very little work explores its capabilities for collaborative activities. This paper describes a case study in which we explored how beginners in Electronics collaborate in order to build logic circuits with our software using an interactive whiteboard as a collaboration platform. Findings of this work point out that interactive whiteboards have demonstrated to be more suitable for collaborative activities than a conventional desktop PC, since they make it possible for students to take turns when interacting with the software. However, this does not always happen and the technology we explored raised issues involving single touch limitations and projection occlusion.

Author Keywords

Interactive board, logic circuits simulator, collaboration, attrition, Boolean logic, digital circuit, freshmen.

ACM Classification Keywords

D.3.3 [Computers and Education]: Computer and Information Science Education— *Computer science education*.

THE COMPUTER ENGINEERING ATTRITION PROBLEM

Like other STEM fields, Computer Engineering undergraduate courses suffer from high attrition rates. Attrition rates for Computer Science (CS) in the U.S. average 19% and can go as high as 60% [1]. Attrition rates are even higher for women [2], what is especially problematic since attracting them to computing related subjects in the first place is already a problem. Women are

outnumbered by men ever since the beginning of the course, in average, by a factor of 1 to 4 in the U.S. [3]. Hence, in order to improve diversity among CS students, how to retain women becomes an issue that needs special attention by the CS teaching community.

This is evidence that it is necessary to rethink how Computing is taught and to try to engage students in different ways. Some initiatives in this direction aim at teaching Computer Engineering skills by having students of all ages developing hands-on projects where they can see the results in the end, like the LEGO Mindstorms Robotics Kit [4], the Lilypad e-textiles kit [5][6] and the Storytelling Alice programming environment [7]. Barker et al. [8] lists amongst approaches to increase the retention of women in Computer Science, increasing student collaboration through pair programming.

In order to tackle this challenge, we are currently structuring at the Pontifical Catholic University a first semester module which aims at engaging students in prototyping and programming activities by having them design and implement circuits as well as program microcontrollers to add behavior to clothes called “Introduction to Engineering: Wearable Computing”. Projects will be executed collaboratively in groups of 3 or 4 students using the Lilypad Arduino kit.

Following what is proposed by [5] and [6], our intention is that by getting in touch with an application that students would not expect to see Computing in, such as fashion, they might be able to broaden their vision of the field and look further for more unusual applications. This might help to fix the image of the computer geek that sits in front of the computer all day that has been reported as being one reason for the high attrition rates in CS [2].

At the same time that it is interesting to have the students participating in hardware and electronics projects since the very beginning of the course, one must consider that students get to university with little or no previous

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Interaction South America 2011, December 1–4, 2011, Belo Horizonte, Minas Gerais, Brazil.
Copyright 2011

knowledge or experience in this area. Therefore, it is necessary to find a way to give them the skills necessary to jump-start the construction process. With this in mind, we developed Blogics!, a logic circuit simulator aimed at beginners in electronics. It brings real world elements into the construction of digital circuits for enhancing their understanding of basic concepts of logic circuits design and implementation.

By taking the approach that collaboration may improve retention, our idea is to use the tool with interactive touch sensitive whiteboards by pairs of students, so they may collaborate and learn together how to build simple interactive systems involving sensors and actuators. This would be one of the first activities of the module to get them ready to get their hands dirty with electronics.

Interactive whiteboards (IWB) have been widely deployed in smart classrooms for both school children and university level students. Using interactive boards and its associated software, teachers are able to drag and drop, hide and reveal, rotate, enlarge and highlight objects, as well as include multimedia content, store and retrieve quickly any content and improve feedback when a particular object is touched [9]. Teachers use interactive whiteboards to have more opportunities to improve a demonstration, the quality of interactions and the way teacher assesses the students; to plan the learning activities and to increase the pace and depth of learning [10]. However, although the students' desire to write on interactive whiteboards is 'overwhelming', investigations suggest that not all teachers involve students in activities that allow them to do so [11]. Changes in role of teachers and students in the classroom facilitated by interactive whiteboards don't seem to be fully exploited [9]. Higgins et al. [12], comparing 184 IWB and non-IWB observed, for example, that IWB lessons had less group work than non-IWB lessons. They suggest that the way the whiteboard holds pupils' attention encouraged teachers in more whole class activities.

Kershner et al. [13] observed that research so far has focused on how interactive whiteboards support teacher-pupil interaction and not on how they can be used for collaborative activities. Hence, they conducted a study that explored whether interactive whiteboards can support young children's collaborative communication and thinking in classroom science activities, concluding that this platform can make identifiable contributions to children's productive communication and thinking. The authors, however, do not consider interactive whiteboards an entirely distinctive or pedagogically transformative learning resource in a primary classroom. Distinctively, our goal is to explore this type of activity with a different target audience, namely beginners in electronics. Hence, we

conducted an exploratory case study comparing how students do it using the software tool with an interactive whiteboard and with a conventional desktop PC. The following sections show our results in this direction.

BOOLEAN ALGEBRA AND LOGIC CIRCUITS!

Developed around 1840 by George Boole and published in 1854 in his book "An Investigation of the Laws of Thought", Boolean Algebra or Boolean Logic is a logic calculus of truth values. It is a two-value algebra, and its values are commonly denoted as 0 and 1, or F (false) and T (true). Besides truth values, it also comprises logic operations. The three basic ones are the conjunction (AND), the disjunction (OR) and the negation or complement (NOT). From these, it is possible to derive other operations such as the exclusive or (XOR), the negation of the conjunction (NAND) and the negation of the disjunction (NOR). Each operation has a corresponding truth table (Table 1) that specifies the outputs for each combination of inputs (Whitesitt, 1961).

These operations can be represented by logic gates. When gates are combined, they form circuits. Boolean Algebra has offered Computer Science a complete system that made it possible to build mathematical models for computational processing. After the invention of transistors and integrated circuits, these operations could be implemented in digital

| | | | | |
|-----------------|----------|----------|----------|----------|
| X | 0 | 0 | 1 | 1 |
| Y | 0 | 1 | 0 | 1 |
| X AND Y | 0 | 0 | 0 | 1 |
| X OR Y | 0 | 1 | 1 | 1 |
| NOT X | 1 | 1 | 0 | 0 |
| X XOR Y | 0 | 1 | 1 | 0 |
| X NAND Y | 1 | 1 | 1 | 0 |
| X NOR Y | 1 | 0 | 0 | 0 |

Table 1. Truth table of some logic functions.

circuits which build the basis of any hardware component. Therefore, it is fundamental that Computer Science and Electronic Engineering students and electronics enthusiasts in general have a solid understanding of how Boolean Algebra and Logic Circuit work.

BLOGICS!

Blogics! is a logic circuit simulation software in which logic gates, wires, sensors and actuators are created and manipulated as moving blocks, as to make the interaction easier for users from different areas other than STEM, such as designers and artists and for beginners in Electronics.

The goal of the system is to offer the basic resources for a didactic logic circuit design experience in physical computing lessons and not a sophisticated simulation of electronic circuits and systems.

Blogics!' paradigm of interaction aims at making students think about the whole system taking into account not only its inputs, its processes and its outputs but also how they interact with the environment (Figure 1). Blogics! contrasts with traditional simulation tools, such as Proteus [14] and LabVIEW [15], that are great for simulating a wide variety of components and integrated circuits, but they can be rather cumbersome for the beginner. Blogics! also differs from many free or low cost digital logic simulators that are designed for educational purposes [16][17], like Digital Works [18], EasySim [19], LogicSim [20] and Logic Circuit Designer [21]. These simulators provide a small and very simple variety of inputs and outputs - switches, LED, 7 bit displays - which is not appropriated to the needs of Physical Computing projects that involves capturing environmental data like temperature, luminosity, humidity and movement and changing this environment through sound, music, vibes and motors.

In its current version Blogics! simulates:

- Logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR. The top, left and bottom sides of the block are used for input and the right side is used for output;
- Truth values: "1", "0", and "-" (blank);
- Sensors working on digital mode: buttons, switches, presence, humidity, luminosity and temperature;
- Environment variables to alter sensor values: finger (button and switch), room with and without people (presence), dry and humid weather (humidity), day and night (luminosity), hot and cold (thermometer);
- Actuators working on digital mode: LED, buzzer, vibration motor, fan;
- Wires.

Simulation occurs in real time and there is no need to press a button to start it. Systems are built by connecting blocks. Each logic gate block has 3 input ports (top, left and bottom sides) and 1 output port (right side). Environment variables influence sensors. When they are connected, the value perceived by the sensor changes accordingly. For example, if a 'Day' block is connected to a 'Luminosity Sensor', the value perceived by the sensor will be 1, but if it is replaced by a 'Night' block, it will be 0.

Blogics!' user interface was conceived to be used with touch sensitive displays. With that in mind, the human

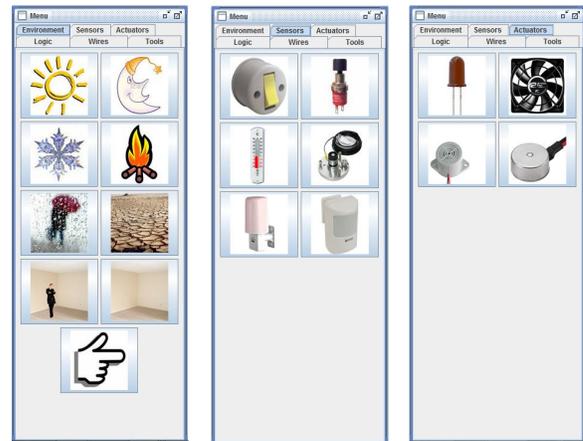


Figure 1. (a) Environmental blocks (b) input blocks (sensors) (c) output blocks (actuators)

computer interaction was designed so that the user only needs to drag and drop blocks to build a circuit. It doesn't require right clicks, double clicks and text entry since these actions can be rather cumbersome in a touch sensitive display [22].

Quigley [23] lists some issues to be considered when developing applications for surface based interaction: the user's stylus, fingers and hands may occlude part of the interface, interface elements may be difficult to select due to the size of stylus, fingers or hands, users may suffer fatigue due to the range of human motion required, the screen can be damaged or dirty, there is a lack of feedback from passive screen surfaces (this doesn't occur with mouse and keyboards because the user can feel the button he pressed), calibration between the display and the sensing elements can become misaligned.

Brown [24] proposed some design guidelines aimed at software for touch sensitive displays that were followed in order to deal with these issues, namely, using touch for intermittent actions (creating and moving blocks), minimizing accidental activation by touching the screen inadvertently, using touch for coarse positioning of blocks and using large icons to increase the touch sensitive area. These design decisions don't prevent the usage with a mouse.

The decision to make it suitable for touch sensitive surfaces was to be possible to organize computer supported collaborative learning activities using an interactive whiteboard. An interactive whiteboard system usually includes a PC that projects images on a wall mounted touch sensitive display. Users interact with the whiteboard by using their fingers or a stylus. Some systems also include a keyboard button that, once pressed, summons a virtual keyboard, where the user can select the appropriate keys with his fingers.

WORKING SCENARIO

Suppose Sarah and Mark, two first year Computer Engineering students just entered our Wearable Computing module. As most of their colleagues, they don't understand how digital circuits work. In order to teach them how they work, Tom, the lecturer, instructed Sarah and Mark to use Blogics! to build a circuit together.

They got up and started Blogics! in a computer equipped with a touch sensitive interactive whiteboard. They wanted to build a circuit in which if it was hot and sunny, it would turn a fan on. Mark clicked on the "AND" button. A block appeared on the screen with the graphic representation of the "AND" gate. Mark wasn't as sure as Sarah as how the gate worked, so he clicked on the "1" and "0" buttons to create two blocks with these truth values and clicked on the "-" to create a blank block. Sarah connected the "0" and "1" input blocks to the left and top sides of the AND block and the "-" output block to the right side. When she connected the "-" block, its value automatically changed to "0". Then, she tried other combinations of inputs: "0" and "0", "1" and "1" and observed the output changes.

"Do you see it, Mark? In order for the output to be "1", both input values must be "1", otherwise it will be "0"." – Sarah observed.

Mark cleared the screen by clicking on the "Tools" tab and on the "New" button. All the blocks disappeared.

"Since we need to know whether it is hot and sunny, we will need sensors. – She said.

"Let's see which sensors we have". – Mark said. He clicked on the "Sensors" tab and they found "Thermometer" and "Luminosity Sensor" blocks to compose their circuit. Sarah selected them and connected the two blocks to the sides of the gate. Then, Mark clicked on the "Actuators" tab and on the "Fan" block. Sarah connected it to the right side of the gate.

"Now we will need to alter the sensors values to see whether our circuit works." – Mark said.

He clicked on the "Environment" tab and created a "Day" and a "Hot" block. When he connected the "Day" to the "Luminosity Sensor" and the "Hot" to the "Thermometer", the "Fan" turned on.

"What will happen if it is sunny and cold?" – asked Sarah.

She promptly created a "Cold" block, disconnected the "Hot" block and connected the "Cold" block to the "Thermometer". The "Fan" turned off.

"What would we need to do if I wanted to turn on the fan even when it is night or cold with a button?" – Tom challenged Sarah.

"That's easy!" – She said. She promptly created a "Button" block and an "OR" block. She connected the "Button" and the output of the circuit they had just made to the inputs of the "OR" block and the "Fan" to its output. She then created the "Finger" block, connected it to the "Button" block. Once the fan was turned off, it turned on and off as Sarah activated the "Button" block. Great!

"You're an expert!" - Said Mark. The final circuit is shown is Figure 2.

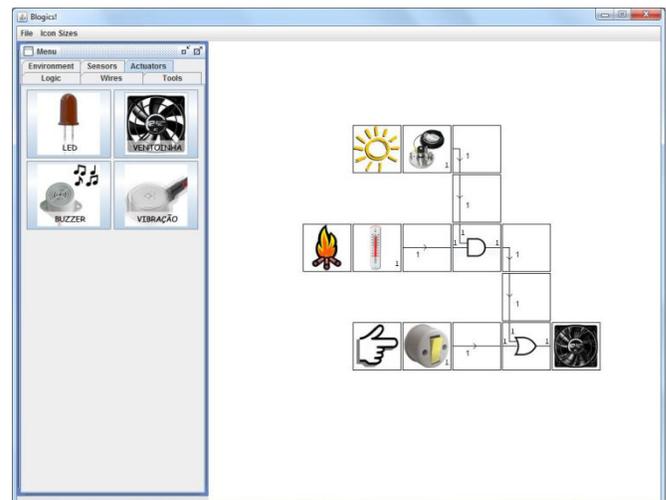


Figure 2. I A circuit built using Blogics! – a fan turns on if it is hot and sunny or if a switch is turned on. The sensors values are determined by the environment variables to which they are connected, namely the "Day" block connected to

RESEARCH METHOD

In order to observe students' interactions with a broad scope, our research question is *how do beginners in electronics build logic circuits collaboratively on an interactive whiteboard?* The case study took place at the Superior School of Industrial Design at the Rio de Janeiro State University. Its objective was to gain insights on the use of interactive whiteboards as a collaboration tool, rather than to prove a hypothesis. Therefore it is classified as an exploratory case study. The participants were all students of a Masters module on Physical Computing, and had little or no knowledge of Logic and Digital Circuits.

The study started with a presentation of Boolean Algebra and logic circuits, followed by a presentation of the software tool, a pre-activity questionnaire, a pre-test, a learning activity using Blogics!, a post-test and, finally, a

post-activity questionnaire. The duration of each activity are listed on Table 1.

The objective of the Boolean algebra and logic circuits presentation (15 min) was to revise the topic and stress some important points, such as what Logic Gates are and how they work, what a truth table is and how to create logic statements with the AND, OR and NOT gates. On a previous lecture of the module, the students were introduced to these concepts, but we wanted them to have the content fresh in their heads. No new concept that was not talked about in the lecture was brought up.

The software presentation (10 min) aimed at introducing the software's interface and main features. Students were shown the contents of the tabs, how to create a simple circuit and how to alter its input values in order to see the output values changing automatically. It was also emphasized that, in order to alter a sensor block's perceived value, it was necessary to connect it to a corresponding

| Activity | Duration (min) |
|-----------------------------|----------------|
| Lecture | 120 |
| Revision | 15 |
| Software Presentation | 10 |
| Pre-Activity Questionnaire | 10 |
| Pre-Test | 10 |
| Learning Activity | 20 |
| Post-test | 10 |
| Post-Activity Questionnaire | 10 |

Table 2. Duration of activities.

environment block.

With the pre-activity questionnaire (10 min) and the pre-test (10 min), we wanted to figure out the students' profiles in order to form groups using the following criteria: students that went well in the pre-test with students that did not, students that had come to the previous lecture with students that had not and students that had experience with touch sensitive surfaces with students that did not have. While the questionnaire covered qualitative questions regarding age, previous knowledge and other personal information, the pre-test consisted of a series of five exercises with gradually greater difficulty. It involved filling truth tables of given functions, designing circuits using AND, OR and NOT gates and one exercise that could be solved using the XOR function, which none of the students were familiar with.

The learning activity (20 min) was the focus of the case study. It consisted of 3 exercises to be solved in pairs, using

either an interactive touch sensitive whiteboard or a conventional PC. We opted for a between-subject study design (in which each participant performs under one condition) rather than a within-subject design (in which all participants perform under all conditions), in order to neutralize learning effects [25]. The class of 8 students was divided into groups of 2: two groups were moved to a room equipped with a SMART Board 680 Interactive Whiteboard [26], a 77" single touch sensitive vertical surface widely used in interactive classrooms, while the other two groups used a desktop PC with standard mouse and keyboard. The members of 3 of the 4 groups had not known each other prior to the activity. In the desktop PC room, two cameras were used: one filmed the screen while the other filmed the students. The Whiteboard room only had one camera, because the display was big enough to appear on film together with the students.

From here on, we use the term platform to refer to either the desktop PC or the interactive whiteboard. The first exercise consisted of a series of instructions that students should follow in order to find out the XOR gate truth table. They were to build the A XOR B circuit using Blogics!, alter the inputs and write down the outputs in order to complete the truth table. Afterwards, they were to add sensors and actuators to the circuit and alter the sensor's perceived value with environment blocks to see the corresponding output values. Students were not allowed to ask any questions to any one apart from their peer during the activity.

In the second exercise, a project specification was given and they had to build the corresponding circuit. No suggestions, gate restrictions or hints were given. We wanted to keep a free pedagogic mechanism so we could explore how the students organized themselves. This problem could be solved using a wide variety of gates, but if the students used the gate they had just simulated, they could find a short solution.

The post-test (10 min) was a list of exercises with similar difficulty level to the pre-test. The exercises had the same structure and the differences were in the used functions or values. As in the pre-test, one of the problems could be solved using the XOR gate. The post-activity questionnaire (10 min) consisted of a series of questions sent by email to be answered using Google Docs Forms, and that was answered online by students within a week. It is important to stress that this was a qualitative study and, therefore, all quantitative data gathered in the questionnaires served for the purpose of supporting our qualitative analysis.

RESULTS

When answering the questionnaire, students commented that doing a Logic exercise with a colleague was "good because one can help the other when he has some doubt about the problems proposed in the exercises."

Collaborative interaction has shown itself to be suitable for individual needs as stated by a student that said: "I think best when I am speaking, so for me it was great." Another student said: "(...) certainly, working in pairs gave me more confidence to solve the exercises."

We also wanted to investigate how they organized themselves to work with the interactive whiteboard. One answer stated: "We weren't concerned about it. The interaction occurred naturally, even though I hadn't known the person that worked with me." Another one stated: "There wasn't an explicit or planned organization. We worked without establishing any method or agreement. It was a spontaneous and tacit agreement." Another one added that "(...) the collaboration was spontaneous within the presented conditions; we did not establish rules, hierarchies, methods."

Some problems reported in the interaction with the platform derived from the fact that the whiteboard being used only supports single touch detection: "(...) some moments of confusion when interacting with the tool (for instance, when we tried to drag different icons at the same time)." Another one said: "Because of the whiteboard's own nature, only one of us could work at a given time. But each one would interfere whenever we felt the need to."

One other problem we noticed in the footage and that students commented on is the virtual barrier created by the projection light. The projector was installed on a table in front of the projection screen, because it is a common setting in Brazilian classrooms and the one adopted by the Rio de Janeiro State University where the study was conducted. Once students started a task, each one positioned himself on one side of the whiteboard, standing out of the way of the projection light. They only invaded the light's space when doing some action on the board and, after this, they would go straight back to their initial position. This created a tacit division of labor, since students would often concentrate on tasks that were to be made closer to their side of the board. The students noticed this problem: "There was also the problem of obstructing the projection when we were to interact with the tool, what is not a problem of the software but of the deployed platform." Another one said: "When both of us were positioned to interact [with the whiteboard] at the same time, there were a lot of shadows."

The members of the first group interacted with the whiteboard by taking turns. They took a trial-and-error strategy and when on error, the other one assumed control of the board. We counted the number of times each one touched the board in each turn and we noticed how balanced the numbers were. On average, one member touched 4.36 times the whiteboard before his peer assumed control and the other one touched it 4.09 times. The standard deviation for the figures was also quite similar:

3.43 for one and 3.68 for the other. The control changed hands 22 times in 20 minutes.

The other group that was also working with the whiteboard had a different interaction dynamics. As previously noted, one member of this group described: "My colleague interacted [with the whiteboard] because he was taller and he could reach the buttons that are higher while I read the problem statement and tried to orient him." In this case, the collaboration scheme derived from physical traits of the students. This effectively happened as observed on the footage. This student did not touch the whiteboard not even once, only pointing to the items that his colleague was to touch. Hence, their collaborative interaction was very similar to the one used when collaborating using a desktop PC. It is worth pointing out that this group only started using the board 10 minutes after the activity started (half of the total time allotted), having spent the first half trying to solve the exercise on paper.

In general, the whiteboard was well praised for the collaborative activity. One of the students stated that "the whiteboard is perfect for this kind of activity." The one that did not directly use the whiteboard for being short said: "I think the benefit is having more room. The conventional desktop PC, with a monitor, seems more inviting to a single user, in a more intimate context. It seems to me that it does not favor a collective participation environment."

The advantages of the desktop commented by the students included: "We are more used to this platform, so we have more agility using it" and "We can solve the exercises while sitting, making it possible for us to make written notes as we used the platform."

When comparing the whiteboard with the conventional desktop computer setups, we observed that the collaboration is organized in rather different ways. One group decided explicitly who was going to control the mouse as soon as the activity started: one student politely asked to be the one holding the mouse and this structure was kept intact for the remainder of the activity. In the other group, one student started with the mouse while the other one told him what to do. This dynamics happened for exactly 3 minutes and 22 seconds and, during this time, the student not holding the mouse physically touched the LCD monitor 11 times trying to tell his colleague what to do. After this, he asked to control the mouse and he did it until the end of the activity. In both groups, the student who held the mouse most of the time was the one on the left, sitting in front of the keyboard.

In both cases, the one that did not hold the mouse did not interfere much, and the collaboration happened by having the one controlling the mouse explaining to the other one what he was doing, rather than asking for suggestions of what should be done. Some comments made by the ones

that did not hold the mouse include “My lack of experience made me accept my colleague’s decisions without questioning the empirical methods he used to solve the problems”. Several responses included negative comments about sharing a desktop computer: “The biggest difficulty was sharing one resource. However, I believe the tool could be used more efficiently with a touch sensitive display” and “Difficulties: Only one person can effectively use the platform at a time, unless the exercise is interrupted in order to take turns.”

Regarding the pos-test activity, we can notice that most students improved their performance, while two kept their scores and one actually got a worse result.

CONCLUSION

Student attrition rates for Computer Engineering, especially for woman, are a very serious problem all around the world. Our approach to solving this is to offer to first semester students an introductory module on Wearable Computing that will engage students in programming and prototyping collaborative activities. In order to bridge the knowledge gap between students’ understanding of logic circuits when they get to university and the minimum requirements to start building actual digital circuits, we developed an educational logic circuit simulator called *Blogics!*, designed for collaborative activities in an interactive whiteboard.

This paper aimed at describing a case study where we explored the collaborative interactions that take place when beginners in electronics build logic circuits together using an interactive whiteboard. The contributions of this work are the qualitative insights on the deployment of interactive whiteboards for collaborative activities, namely the interaction dynamics, the limitations of the interactive whiteboard as a collaboration tool and some design guidelines for applications.

We observed two types of interaction dynamics: the one where the students solved the exercise by taking turns when using the whiteboard and the one where one student used the whiteboard while the other dictated the actions to be taken.

By comparing the use of the whiteboard with the use of a conventional desktop PC, evidence shows that the whiteboard is more suitable for collaborative activities since it gives more room for students to move and it eliminates the awkwardness of asking for the mouse. However, these new opportunities are not sufficient to foster a balanced peer to peer collaboration if students decide to adopt a social protocol with different roles, such as one in charge of operating the hardware and the other conducting the activities.

Two issues were cause of confusion while collaborating: the single touch limitation and the virtual barrier created by the projection light. The single touch problem was circumvented by the students with their collaboration protocol: either only one student interacted with the board or they took turns. Therefore, we suggest that a multi touch vertical surface would be better suited for a collaborative activity such as this than a single touch one.

Other aspect that must be considered for teachers who want to conduct collaborative learning activities with interactive whiteboards is the physical setting of the classroom. It is important to make sure that all students can reach all points of the whiteboard, either by lowering the board or by providing blocks where the students can stand. Also, application designers might want to make menu items available at the bottom of the screen or in a configurable position.

The problem of the virtual barrier is inherent to front-projected displays. A person’s body or hand may occlude the display or block the line of sight for a camera system. This may affect the collaboration dynamics by making it difficult for students to move from one side of the whiteboard to the other. This problem can be mitigated by using a short throw projector, in which the projector can be positioned close to the screen; a rear-projected display, in which the projection comes from behind the display; or a self-projected display, such as LCD or plasma screens [23].

Based on the observed results, future work will consist of tailoring collaborative activities that take into account the interaction dynamics around an interactive whiteboard to best take profit from the work done together by the students in order to fast forward electronics activities.

ACKNOWLEDGMENTS

Hugo Fuks is the recipient of individual grants awarded by the CNPq and the Rio de Janeiro State Research Council (FAPERJ) Project entitled “Cientista do Nosso Estado.” We would like to thank Prof. Fernando Reiszal Pereira, lecturer of the Physical Computing module at the Superior School of Industrial Design at the Rio de Janeiro State University for the support in making this case study happen.

REFERENCES

1. Sloan R. H., Troy P. 2008. CS 0.5: a better approach to introductory computer science for majors. In: 39th SIGCSE Technical Symposium on Computer Science Education (Portland, OR, USA, , pp. 271-275, ACM, New York, NY
2. Margolis J. and Fisher A. 2002. *Unlocking the Clubhouse: Women in Computing*. The MIT Press, Cambridge, MA

3. Hill C., Corbett C., St. Rose A. 2010. Why so few? Women in Science, Technology, Engineering, and Mathematics. AAUW Research Report, Washington, DC
4. Ringwood J. V., Monaghan K., Maloco J. 2005. Teaching Engineering Design through LEGO Mindstorms. *European Journal of Engineering Education*, Vol. 30, No.1, pp. 91-94.
5. Buechley L., Eisenberg M., Catchen J., Crockett A. 2008. The LilyPad Arduino: Using Computational Textiles to Investigate Engagement. In: SIGCHI Conference on Human Factors in Computing Systems (CHI '08), Florence, Italy, pp. 423-432
6. Lau W. W. Y., Ngai G., Chan S. C. F., Cheung J. C. Y. 2009. Learning programming through fashion and design: A pilot summer course in wearable computing for middle school students. In: Proceedings of SIGCSE '09, Chattanooga, Tennessee
7. Kelleher C., Pausch R. 2007. Using Storytelling to Motivate Programming. *Communications of the ACM*, 50, 7, pp. 59-64
8. Barker, L., Garvin-Doxas, K., Roberts, E. 2005. What can Computer Science Learn from a Fine Arts Approach to Learning. In: Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, St. Louis, Missouri
9. Kennewell, S. 2006. Reflections on the Interactive Whiteboard phenomenon: Synthesis of research from the UK, AARE Conference, Adelaide, November
10. Becta 2004. Getting the most from your interactive whiteboard: a guide for secondary schools Coventry, Becta ICT Advice <https://www.education.gov.uk/publications/eOrderingDownload/15090.pdf>, last accessed on 8 July, 2011
11. Smith, H. J., Higgins, S., Wall, K. Miller, Jen 2005. Interactive whiteboards: boon or bandwagon? A critical review of the literature. *Journal of Computer Assisted Learning* 21, Blackwell Publishing, pp 91–101
12. Higgins, S., Falzon, C., Hall, I., Moseley, D., Smith, F., Smith, H. and Wall, K. 2005. Embedding ICT In The Literacy And Numeracy Strategies: Final Report, Newcastle University, Newcastle, UK.
13. Kershner, R., Mercer, N., Warwick, P., Staarman, J. K.: 2010. Can the interactive whiteboard support young children's collaborative communication and thinking in classroom science activities?. In: *ijCSCL* 5 (4), pp. 1-25, Springer
14. Proteus VSM Simulator. <http://www.labcenter.co.uk> , last accessed 3 March, 2011.
15. NI LabVIEW. <http://www.ni.com/labview>, last accessed 3 March, 2011.
16. Burch, C. 2002. Logisim: A Graphical System for Logic Circuit Design and Simulation *Journal of Educational and Resources in Computing*, Vol. 2, No. 1, March
17. Wolffe, G., Yurcik, W., Osborne, H. and Holliday, M. 2002. Teaching computer organization with limited resources using simulators. In SIGCSE Technical Symposium on Computer Science Education, February 27- March 3, Covington, Kentucky, USA, 2002, pp 176-180
18. Digital Works. <http://www.spsu.edu/cs/faculty/bbrown/circuits/howto.html>, last accessed 3 March, 2011.
19. EasySim, <http://www.research-systems.com/easysim/easysim.htm>, last accessed 3 March, 2011.
20. LogicSim, http://www.tetzl.de/java_logic_simulator.html, last accessed 3 March, 2011.
21. Logic Circuit Designer http://download.cnet.com/Logic-Circuit-Designer/3000-2054_4-10840569.html, last accessed 3 March, 2011.
22. Rekimoto J. 1998. A multiple-device approach for supporting whiteboard-based interactions. In: Proceedings of CHI'98.
23. Quigley A. 2009. User Interfaces for Ubiquitous Computing. Book Chapter in *Ubiquitous Computing Fundamentals*, Chapman & Hall/CRC.
24. Brown C. 1998. M.: *Human-Computer Design Guidelines*. United Kingdom: Intellect, pp. 135-137
25. Cairns, P., Cox, A. 2008. *Research Methods for Human-Computer Interaction*. Cambridge University Press, Cambridge UK,.
26. Smart Board, available at www.smarttech.com, last accessed 8 July, 2011