The GoGo Board: Moving towards highly available computational tools in learning environments.

Arnan Sipitakiat[1], Paulo Blikstein[2], David P. Cavallo[3]

Massachusetts Institute of Technology

Keywords: *Technologies for learning, Systemic analysis, low-cost technology, data-sensing, control.*

Abstract:

This paper presents a new framework to tackle the lack of technology availability in learning environments such as schools. The framework is based on the hypothesis that the commonly heard reasons for the scarcity of technology, such as high-cost and complicated budgeting models, are side effects of centralized management styles and an exaggerated belief in the mass-production mindset. The new framework emphasizes communities as producers of their own tools. The GoGo board, a computer-interfacing micro-controller board, is presented as an instantiation of the framework. Preliminary observations of the GoGo board usage and assembly by schoolteachers in Brazil are presented.

1.0 Introduction

Availability of learning equipment, materials and infrastructure has always been a crucial challenge in educational systems all around the world, especially in developing countries. Before even considering the purchase of computers, public administrators have to pay attention to much more basic problems such as lack of books, paper and desks, as well as fixing leaking roofs, raise the salaries of teachers and build new schools. This fact poses a serious paradox when considering the introduction of any technology project. *"The computers and the technology are good, but we cannot afford them"* is the typical argument. We believe this situation is a result of an unrecognized mindset governing how people think about technology availability. This paper analyses this situation from a different perspective and presents an alternative framework that we believe would lead to learning environments where the issue of technology availability becomes part of a fruitful learning process rather than a stumbling block.

2.0 Roots of the crisis: a systemic analysis

2.1 Side effects of the mass-production mindset

This paper develops a framework treating technology availability as more than merely a cost issue, but as part of a much more complex system; shaped by the economic model of our society and the rigidity of our learning institutions.

We live in a society that favors higher outputs, lower costs, and better equity though standardization. Fruits in supermarkets can come from the other side of the globe while having a reasonable price. Newspapers contain hundreds of articles but their cost is very low. This is possible because those commodities are mass-produced. However, the mass-production system often breaks down and creates undesirable side effects. Two examples follow:

- **Cost** – Some commodities have not been as cheap as they could. The personal computer is a good example. The primary reason behind this upholding is that the computer industry has been holding the price by offering technologically advancements that have been driven by its own commercial strategies, and not necessarily by the need of consumers. Any attempt to go against this paradigm[1] would face the rapid growth of user expectations set by the opposing paradigm.

- **Obsolescence effect** – This effect is pushed to the extreme in the computer industry. As computers rapidly advance, the software they run increasingly requires more computing power as well. As a result, most computers become obsolete within a few years simply because they cannot run the newer operating systems and software. Spare parts also become scarce. Thus, maintenance becomes expensive or simply impossible.

Institutions such as education, health care, and transportation have adopted the industrial mode of production. Every aspect of the institution becomes standardized, including equipment purchase procedures. Ivan Illich describes that this industrial mode of production appears favorable at first. But over time they will quickly pass a critical point that will begin to create destructive side effects, or even work against its initial goals [1]. The following side effects are common in the purchasing process of school institutions.

- **Funding Model** – Budgeting and funding in large institutions such as schools are highly standardized. Capital equipment purchase must be well planned in advance. Vendors compete by offering the lowest price, product guarantee, customer support, and training. Once this process becomes established, it becomes expected. Other spending schemes that are different become undesirable or even unacceptable. This situation is called “radical monopoly” [1]. An undesirable side effect of this system is its long turnaround time.

[1] One example is the low-cost Linux terminal manufactured by Samurai, a Brazilian technology company.

Schools loose the flexibility to purchase equipment "as needed" at a particular time and situation. Everything must be planned ahead. Also, marketing strategies used by the vendors often take advantage of the lack of technical and pedagogical background of the schools administrators [2].

- **Consumer Mindset** – Because of the rigid funding model, other modes of acquiring tools (such as the one proposed in this paper) becomes extremely difficult. Thus, people are "over programmed" to become mere consumers of products [1]. In recent years, purchasing computers is to buy a ready-to-use machine with the right software package, and with the right training courses. Fixing a computer means calling tech-support or to send it to a repair shop. The more one becomes a consumer - the more one has to pay. This mindset inevitably reinforces the price issue.

- **Optimization** – In a system where buying more means cheaper unit price, top level managers of institutions often maximize their budget by making one large purchase and then distribute equipment equally to each department. It is not uncommon for a municipal secretariat of education to purchase thousands of computers and then distribute them to the schools. When this optimization exceeds a certain point, schools become merely receivers of whatever is decided for them. This is more harmful than it seems. It prevents any other alternatives from taking place even if the school and the teachers are willing to try, as we observed in our case studies in São Paulo and Curitiba, Brazil.

These side effects are often unrealized and people take the consequences as a given fact. For these people, any viable solution mush fit within this artificial, but yet extremely rigid, system. So, the only solution seems to be for schools to receive higher budget to buy more expensive technologies. This solution of course never happens or is extremely rare. On the surface, it seems like there is no solution to this problem. But once we analyze its roots, we start to envision some possible new strategies.

2.2 Story from a Fish Farm in Thailand

This story demonstrates a concrete example of an extremely useful technology being rejected because of its impracticality. Lek was a fourteen-year-old student at Tongtip School, located in a rural area of northern Thailand [3]. In January 2001 she participated in a five-week after-school program working with computers and LEGO RCX programmable bricks. Towards the end of the program, she designed a device with the RCX to control insect lights at her fish farm located approximately one kilometer from her house. Lights at the fish farm attracts insects which would fall into the pond and become food for the fish. Most insects come out only for a few hours after sunset. Lek's goal was to avoid walking to the fish farm to turn on and off the lights.

Figure 1: The fish farm (left) and the RCX controlled light switch (right)

Lek designed a sophisticated system using a light sensor and a relay. She programmed the RCX so that when the sun sets, the relay would switch the insect light on. The RCX would then wait for a few hours before switching the lights off. She went through an extremely powerful learning process to accomplish her goal. She was involved with ideas in fields of electronics, computer programming, and physics[2] . However, Lek's father was concerned about his daughter taking responsibility for a US$ 120 RCX Brick[3]. Although he acknowledged his daughter's project as something extraordinary, he demanded Lek to return the brick as soon as possible.

It was evident that the tension that happened was the main stumbling block for Lek. The schoolteachers had the same concern about using these tools that almost costs more than their monthly salary. To the school, the only way they could obtain RCX bricks is to set aside some money from their annual budget. But the school's budget is merely enough to purchase essential supplies. Even worse, it would be difficult to justify their decisions to the municipal level managers who decide their budget. The school could easily loose that portion of their budget to other schools that purpose for more fundamental equipment (e.g. books, tables, chairs). Thus, although every teacher would agree that the RCX is a powerful tool for learning, the practicality of using the RCX in the school is seriously doubted. The same issues about cost of technology were observed in our fieldwork in São Paulo, Brazil [2].

3.0 A new approach

This paper proposes a new strategy that is developed based on the realization of the institutional and mindset problems. It is important to clarify that we do not develop or propose strategies that simply reject everything about the mindset we are criticizing[4]. The goal is to set people's mind free from the constraints they often take as a given fact, and create space for new ideas. This work is based on the perspective that the current school system and mindset has become off-balanced. They favor one idea to an extreme and is suffering from it. We believe in systems that utilize multiple philosophies and treat equilibrium as a dynamic process.

---

[2] See Arnan Sipitakiat's thesis for detailed information [3]

[3] http://www.mindstorms.com

[4] John Dewey calls such assumptions the *Either-Or* phenomenon [8]

Our framework suggests that people become producers of technology rather than being merely consumers. The goal is to create a culture that nourishes the idea of "making one's own tools." Our hypothesis is that this culture will transform technology availability into a different and much more manageable issue. Of course there are some tools that are impossible for people to build or that would end up being too expensive or would lack technical support. However, the availability of new technology for personal fabrication[5], as well as the relatively high availability of electronic and computational technologies are rapidly changing this possibility. The GoGo-board framework presented in this paper benefits from powerful and low-cost micro-controllers that have been recently available.

The introduction of new ideas in institutions such as schools typically faces resistances. The system tends to push the new ideas back to fit into the existing structure and maintain its equilibrium [4]. This resistance normally happens when the new idea is delivered to the system in an imposing manner. Thus, in this work we emphasize the spread of ideas using an alternative model. We aim to identify and support a small group of *enthusiast* who are attracted to this new framework; people who work with the **GoGo board** because they want to not because they are obliged to. In the case of schools, group members can span between schools and members can work together both physically and remotely. The work of these enthusiasts will become a subculture striving within the existing system, which hopefully will slowly gain momentum. This biological approach allows the framework to mature and expand at an appropriate pace.

4.0 Implementation

This project instantiates the proposed framework by concentrating on the GoGo board, a small electronics board that belongs to the "Programmable Brick" family (such as the commercially available LEGO Mindstorms RCX brick[6] and the Cricket[7]). Thus, the usability of the GoGo board builds on more than a decade of research that proves how beneficial it can be in learning environments [5, 6, 7]. The GoGo board is a suitable fit for our theoretical framework for many reasons:

- It is an alternative to the much more expensive commercial devices, especially the RCX brick from LEGO, which costs approximately US$ 120 in the United States. The GoGo board costs less than US$20 [8].
- The components required to build such device, particularly the micro-controllers, has recently become widely available at a reasonable price in different countries. The micro-controller used on the GoGo board costs approximately US$5.00, even in Brazil, with all import and local taxes included.
- PCB (Printed Circuit Board) assembly is straightforward. Anyone with basic soldering skills can put together a GoGo board.
- The GoGo board design can be simple but yet it can be used in a wide variety of sensing and control projects. There is a vast space for creativity, which

[5] Such as laser cutters, electronic circuit printers, powerful software development tools, electronic sensors and logic components.
[6] See http://www.mindstorms.com
[7] See Gleason Research at http://handyboard.com/cricket.
[8] When parts are purchased in Brazil

opens up many windows for people to adapt and redesign the board for their particular interest.

4.1 GoGo Board Overview

The GoGo board basically provides a channel for interaction between the physical world and the on-screen digital world. Users can connect various sensors to the board and program the computer to interact with the sensor values. It also has output channels that allow users to control devices such as motors, small lamps, LEDs, and relays. Its features are similar to other widely available computer interface hardware. However, it is designed to work well with LCSI Microworlds Logo, a powerful, media rich version of the famous Logo programming language. Thus, it offers a rich environment with which users can work. The GoGo board also has libraries for other environments including C++, Visual Studio (Visual Basic, Visual C++), and Microsoft Office (Excel, Access, PowerPoint, Word). Libraries for other programming environments such as Java and Squeak will be developed in the future.

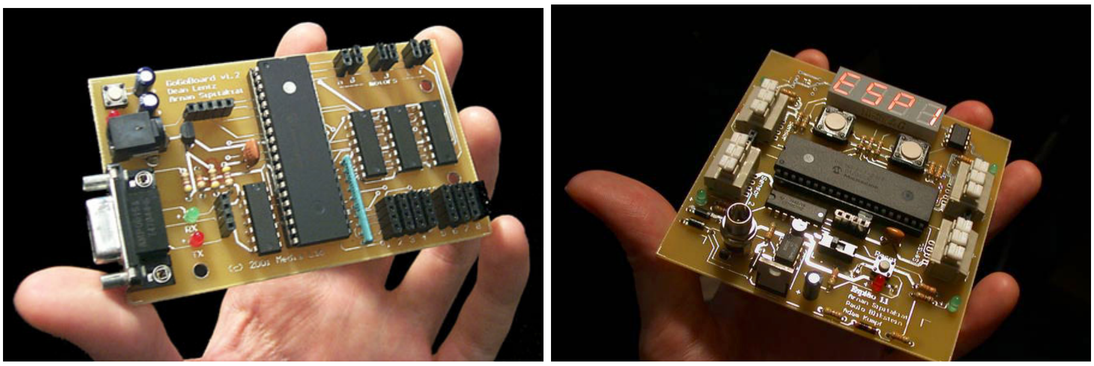

Figure 2: The GoGo (left) and the Espion boards (right).

The GoGo board is designed to be tethered to the computer. The computer tells the GoGo board what to do and the board responds accordingly. This is its main difference from the programmable bricks, where programs are downloaded to the device and then ran on the device independently from the host computer. We are currently developing another version of the board, the *Espion*, which runs autonomously and thus is designed to behave like other programmable bricks.

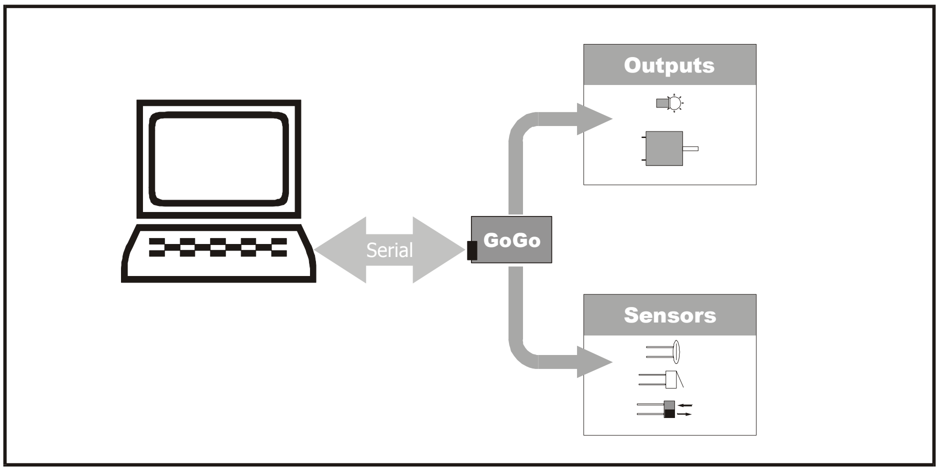

Figure 3: The GoGo board is connected to a computer via a serial cable and acts as an I/O board reading sensor values and controlling output devices.

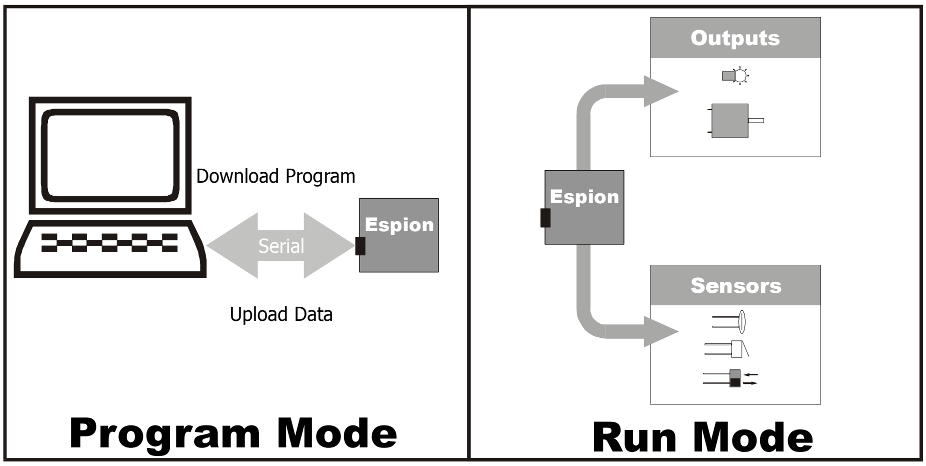

Figure 4: The Espion board can be programmed by the computer (left) and later run independently (right).

An example application of the GoGo board is computer game design and interfacing where learners create various game controls in the physical world to interact with games written on the computer. The following example shows a race game where the two runners on the screen are controlled by a simple step sensor made out of ice-cream sticks and kitchen aluminum foil.

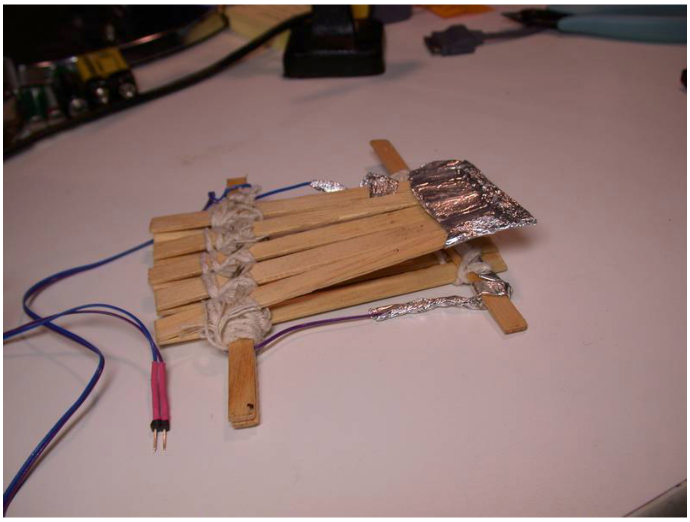

Figure 5: Step sensor

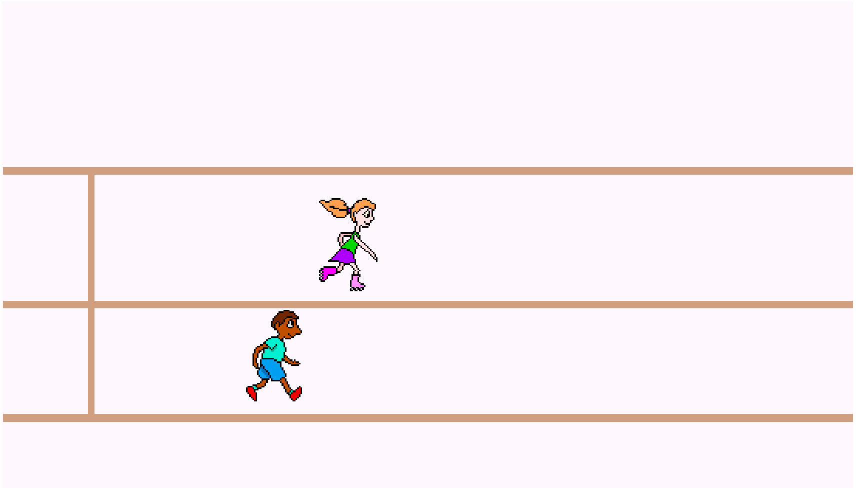

Figure 6: Race game using the GoGo board as an input device.

```
Forever [
     If sensor1 < 500    [
                    Forward 10
                    Waituntil [sensor1 > 500]
                    ]
     ]
```

The above code is a simple Logo program that controls a runner. **Sensor1** is a primitive added to Logo by the GoGo library. When no sensor is connected, **sensor1** will return 1023 [9]. On the other hand, if the sensor port is short-circuited (i.e. by two connecting wires) it will return zero. In this example, when the step sensor is activated the sensor value should be very low, if not zero. Thus, "**if sensor1 < 500**" simply means "if step sensor is activated." With this simple idea in mind, the code above moves the runner forward every time a player steps on the sensor. It waits until the sensor is released before moving the runner again to force the player to release his or her foot from the sensor. If a second step sensor is connected to **sensor2**, a similar code can control the second runner. Thus, we have a game that two players compete by stepping on the sensor as fast as possible.

The following are some example of projects that have been done with the GoGo board.

Intelligent Bathroom

This group wanted to create a model of their school's bathroom that would detect whether or not a person flushes the toilet after he or she uses the bathroom. If the person does not flush, a voice message will remind that person to do so. This is a perfect project for the GoGo board because they want to interact with the physical world and play a voice recording, which is something hard to do outside of a computer.

[9] The ADC resolution is 10 bits

The group created a model of the bathroom out of styrofoam. They built a Lego robot to represent a person using the bathroom. Then, they made infrared sensors[10] to detect when the Lego robot enters and leave the bathroom. They attached a touch sensor to the toilet. Whenever someone flushes the toilet, the touch sensor would be activated. Finally, they digitally recorded the warning message onto the computer. All the programming, which processes the all the sensor inputs and controls the audio playback, was done in Logo.

Dance-Dance-Revolution

This group wanted to create a game and as everyone like to dance samba, they decided to make a Dance-Dance-Revolution game, in which the user has to step on four pads on the floor following the instructions on the screen. This is again a perfect project for the GoGo board, as it needs physical inputs and needs the kind of processing and display that is difficult to accomplish without using the computer. The group created four step-on sensors and connected them to the board. A computer program is then created in Logo to detect how well the player is dancing and award the player with points accordingly.

Figure 7: The intelligent bathroom project (left) and the dance-dance revolution game (right).

The Calorie Scale

This group wanted to improve people's awareness of how much calories there are in the food they eat. The idea was to create a scale to measures the weight of the food, then convert the weight into calories. The scale was attached to a bend sensor. Thus, the more the food weighs, the more the sensor bends. Then, a Logo program coverts the sensor reading into weight and finally into calories. A set of buttons was created for the user to select which type of food they were weighing (rice, meat, corn, etc). The calories value was then printed on the computer's screen.

[10] Infrared sensors can detect the presence of an object. This is the same idea of the sensors used in automatic doors.

Power Friendly House

Students built a model house designed to save energy. The four bedrooms had light bulbs and fans, which were wired to the GoGo Board. The students drew the floor plan of the house in the Microworlds environment and added buttons that would control and time all the appliances of the house, turning them off automatically to save energy.

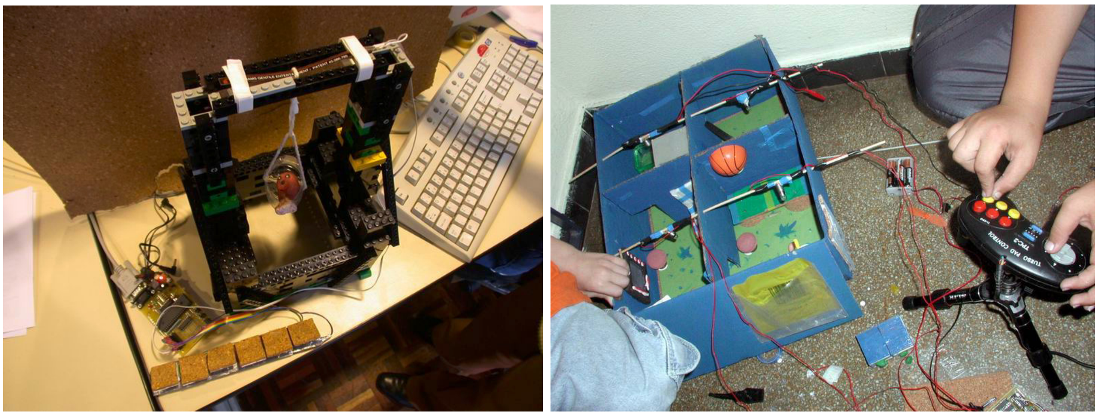

Figure 8: The calorie scale (left) and the power friendly house (right)

4.2 GoGo board Design Goals

Although the technical features of the GoGo board are meaningful, they are not the main issue of its framework. The following are design goals that reflect the deeper aspects of this framework.

4.2.1 Designed for users to build

We do not expect all users to build their own boards. However, we observed that some are indeed interested in building them. If we want users to make GoGo boards, it must be possible for them to do so. This issue involves many factors. First, the parts that are used should be easy to find and at reasonable prices. The only complex component on the GoGo board is the micro-controller unit (MCU). The GoGo board uses a *Microchip PIC* MCU[11], which are among the most popular and available MCUs in the market. Other components on the board are simple and common, such as transistors, resistors, and Light Emitting Diodes (LEDs). These parts are commonly used in electronics repair shops and are relatively inexpensive. In the United States these parts can be easily purchased through online retailers or at Radio Shack. In many other countries, such as Brazil and Thailand, parts are available at "electronics districts" commonly found in large cities. The following table shows the availability and price of components needed to assemble one GoGo board in Brazil, based on our field research in August 2002.

---

[11] See http://www.microchip.com for more information.

Part	Part Price R$	US$*	Units per Board	Total R$	US$
Microcontroller	16.306	5.260	1	16.306	5.260
DB9 Serial Connector	1.176	0.379	1	1.176	0.379
Power Jack	0.750	0.242	1	0.750	0.242
Red LED	0.107	0.034	2	0.214	0.069
Green LED	0.116	0.038	1	0.116	0.038
Resistor 4.7K Ohm	0.013	0.004	1	0.013	0.004
Resistor 470 Ohm	0.013	0.004	3	0.039	0.013
Resistor Pack 10K	0.570	0.184	1	0.570	0.184
Female SIP Connector 2x2	0.290	0.094	3	0.870	0.281
Female SIP Connector 2x8	1.162	0.375	1	1.162	0.375
Female SIP Connector 1x8	0.799	0.258	1	0.799	0.258
On/Off switch	0.500	0.161	1	0.500	0.161
Crystal	1.493	0.482	1	1.493	0.482
Hex Inverter	0.550	0.177	1	0.550	0.177
Capacitor	0.100	0.100	1	0.310	0.100
Push Button	0.150	0.048	1	0.150	0.048
5V Power Regulator	0.742	0.239	1	0.742	0.239
IC Socket	1.617	0.522	1	1.617	0.522
Serial Cable	7.000	2.258	1	7.000	2.258
Power adaptor	7.000	2.258	1	7.000	2.258
Total				**41.38**	**13.35**

*Currency exchange rate: US$ 1 = R$ 3.10

The GoGo board also uses parts that are easy to solder. There are no surface-mount components. This selection makes it possible for people with no or little soldering experience to put together their own GoGo board.

Figure 9: An electronics store in Sao Paulo, Brazil

Would people be interested in making their own boards? This is a fundamental question of the GoGo framework. Our preliminary experiences working with schoolteachers in and in Curitiba, Brazil[12] show that schoolteachers, both men and women, are open to this idea. We have engaged people to solder wires and make sensors for the GoGo board. Soldering was new to most people but it did not appear to be a problem.

[12] Observed during a two-week workshop in July 2002. See *http://ii.media.mit.edu* for more information about the workshop.

Though we have not gotten to the stage where we would encourage people to make their own GoGo boards, we already have one successful case. During a workshop in July 2002 in Curitiba, Brazil, Jordana, a primary school teacher, was intrigued when she heard during a GoGo board demonstration that she could make her own board. She decided to put together a GoGo board to use in her project during the workshop. She had never used a soldering iron before and does not have any electronics background. However, she mentioned that her husband often fixes electronic appliances. That is where she gained her familiarity with the idea of soldering and making electronic devices.

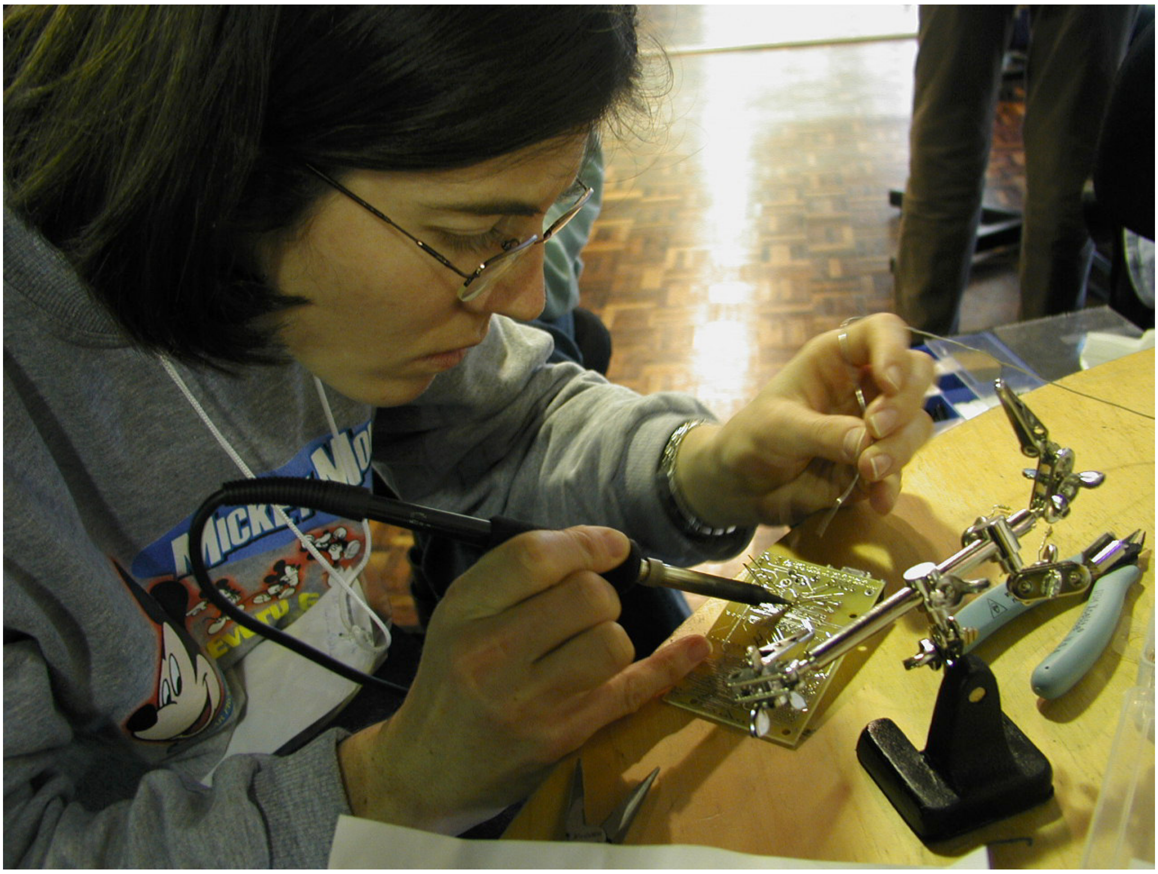

Figure 10: Jordana assembling a GoGo board.

Jordana is an example of the people we wish to identify; people who want to make tools for themselves and have fun and pride doing so. These people can form a group that learns about the GoGo board and help lead others who may later become interested.

4.2.2 Designed to be redesigned

It is a misconception to think that the GoGo board is designed to hide the technical details from the user and make it easy and cheap to build. A culture of “making your own tool” that merely assemble pre-designed devices misses the point. For people to move beyond being mere consumers to become active producers, it is at most important to engage the people in thinking about the design of their tools. The GoGo framework aims to encourage this process.

Design can happen in many levels. It can mean changing components (i.e. connectors, switches, or the casing), reconfiguring the features of the board (i.e. the number of inputs and outputs), or adding new features (i.e. adding memory, a display, or a battery charger). The process needed for these design changes is normally complex and requires good background knowledge in electronics. The GoGo board, on the other hand, provides a solid core hardware system for people to design upon. The GoGo board hardware contains the fundamental components required for a basic MCU based system. These components include the power supply, serial communication with the computer, a bus system for expansion, sensor inputs, and output ports. This allows non-technical people to quickly get involved in the design process without the need for extensive training in electronics. However, this simplicity does not mean the GoGo board lacks complexity and functionality. The more complex electronics do exist but the users do not have to understand them as a prerequisite to be able to adapt the board.

The GoGo board also provides a wide range of software development kits (SDK) for people to easily access the features of the GoGo board from the computer. The available SDKs include GoGo Active-X control, Microworlds Logo Library, C++ Class Library, and WinAPI DLL. The micro-controller contains a generic program module that provides users with most fundamental functionalities, such as serial communication, sensor readings, and output port controls. Thus, users have solid ground to develop upon both on the computer and on the GoGo board micro-controller.

5.0 Future Directions

This work is still in its early stage. Though the theoretical framework and implementation model are in place, much research and fieldwork are still needed to prove the concept. The following are components of the framework that are being implemented in Brazil.

5.1 Establish a group of GoGo board enthusiast

Though the GoGo board is designed to be constructed by the user, it would be unrealistic to believe every user would want to make their own boards. A better model is to establish a group of people who are truly interested in the GoGo boards and want to build and adapt the board. This group can act as the designers of the GoGo board, which could later be produced and distributed to schools. This group could comprise of schoolteachers, technical persons from universities, or students. This group of people then communicates using on-line tools, such as web boards, or mailing lists. This method is similar to open-source projects such as Linux, Apache, MySQL, etc.

5.2 Create a GoGo board design environment

The design tools necessary for GoGo board modifications should be freely available and easy to access by everyone. These tools include PCB design and micro-controller programming environments. The current tools used to design the GoGo board are commercial products that are normally available only at technical institutions. Thus, it is still rather complicated to allow every user access to the technical development of the board.

6.0 Conclusions

The GoGo framework is an attempt to break the centralized and mass-production mindset for technology projects in schools. Instead of being just consumers of end products, educators and students have the possibility to work as a group to design their own tools. Instead of buying expensive imported equipment, schools can assemble or contract local companies to produce the hardware. By establishing a relationship with local experts in universities or companies, educators can find support and discuss improvements to the GoGo board that would better fit their needs. Our fieldwork demonstrated that teachers do get interested in building and using the GoGo board. They were surprised at how low-cost the board can be as well as how the components are available locally. We believe that the observations presented in this paper hint that a new and better model of technology acquisition would be greatly appreciated and its possibility is not so far fetch as it may seem.

References:

[1] Illich, E. (1973). *Tools for Conviviality.* New York: Harper Colophon Books.

[2] Blikstein, P. (2002). *The Trojan Horse as a Trojan Horse: impacting the ecology of the learning atmosphere.* Cambridge, MA: MIT Media Laboratory Master's Thesis, Massachusetts Institute of Technology.

[3] Sipitakiat, A. (2001). *Digital Technology for Conviviality: Making the Most of Students' Energy and Imagination in Learning Environments.* Cambridge, MA: MIT Media Laboratory Master's Thesis, Massachusetts Institute of Technology.

[4] Tyack, D., & Cuban, L. (1995). *Tinkering Toward Utopia: A Century of Public School Reform.* Cambridge: Harvard University Press.

[5] Martin, F., Mikhak, B., and Silverman, B. (2000). MetaCricket: A designer's kit for making computational devices. *IBM Systems Journal*, vol. 39, no. 3-4, pp. 795-815.

[6] Cavallo, D. (1996). *Leveraging Learning through Technological Fluency.* Cambridge, MA: MIT Media Laboratory Master's Thesis, Massachusetts Institute of Technology.

[7] Resnick, M., Martin, F., Sargent, R., and Silverman, B. (1996). Programmable Bricks: Toys to Think With. *IBM Systems Journal*, vol. 35, no. 3-4, pp. 443-452.

[8] Dewey, J. (1938). *Experience & Education.* New York: Collier Books.

Authors:

Arnan Sipitakiat, Research Assistant
Massachusetts Institute of Technology, Media Arts and Sciences
E15-320H, 20 Ames St.
Cambridge, MA
arnans@media.mit.edu

Paulo Blikstein, Research Assistant
Massachusetts Institute of Technology, Media Arts and Sciences
E15-320H, 20 Ames St.
Cambridge, MA
paulo@media.mit.edu

David P. Cavallo, Research Scientist
Massachusetts Institute of Technology, Media Arts and Sciences
20 Ames St.
Cambridge, MA
cavallo@media.mit.edu