What can studio-art have to do with the engineering curriculum?

Authors:

Etienne Delacroix, Facultad de Ingenieria UDELAR , Instituto de ingenieria Electrica, Julio Herrera y Reissig 565,Montevideo 1130 ,Uruguay, eti@mit.edu

Abstract — *Combining aspe cts of engineering with traditions of studio art we investigate an interface between both worlds: using an a ccumulation of electrodigital refuse, taken as a "raw expressive medium", an elective course (TAP: "taller de arte y programacion" – Studio of art and programing") takes a large mixed group of students (engineering, art, architecture, music, etc..) with very different levels of skills, for a sustained immersion into an exploration context. Students deconstruct, reconstruct funcional hardware and use programming to produce a documentation of the process*

Index Terms — *Interfaces, Art, Progra mming, Electronics, Low -cost.*

1. Background

The work presented here is the outcome of somewhat unusual circumstances and results from a long process of experimentation in the context of a "traditional" artist studio but with a previous background in physics. While during my dozen years of studying and practicing research in Physics, starting in the late sixites, i had maintained a strong interest in art and music, it is only at the onset of my second post-doc that i came to the conclusion that these years of doing science should be taken as a foundation for doing art.

Followed 15 years of studying and practicing drawing and painting in the most traditional context: an artist studio in a rundown building in the center of Paris, completely cut-off from academic circles, the inevitable consequence of my successive transplants between Belgium, the United States and France. An immediate legacy of physics in my work as a painter, however, was that i was aware of computers, of the gestures to assemble them, of the process of programming them, of the way they could be used to model and control physical processes. Learning to paint went thus on par with learning how to use the, by then, emerging microcomputers, starting from assembling and soldering them from the bare electronic parts for Z80based, CP/M machines coming in mail order kits, and then learning how to program them in an array of diverse languages (Basic, asembly, lisp, C,forth). Not an easy task: This implied a culture that was completely absent from the art context in which i was living and working. It took two years of work to see a first graphic pixel and and then the luck to stumble upon the right articles in "Dr Dobbs journal" and upon the first edition of Ableson and Sussman's book "the structure of computer programs" to gather the tools to initiate a usable developpement. During this time i kept focusing on one question: What would happen to the practice of a traditional painter, if he was given "free" access to computers in all its aspects, in the same way that a studio artist must have unrestrained acess to the "totality" its chosen media.

The answer came over the years as a series of drawings and paintings deeply woven with a corresponding series of sofware "sketches" which had evolved into singular, full fledged producion tools. Using computers had allowed me to organize my work as sequences of gestures which would lead to paintings. Style had evolved into a generative process based on combining clusters of elementary shapes from a reduced set i called "morphems", following a complex pattern of feedback between painting and writing code. From the early nineties on, the use of multiple machines and crude networking and database technologies allowed me to make physical installations and performances in which a painterly, narrative, interactive process had begun moving onto the stage. I had been working alone with very little access to competent programmers or intellectual resources from universities.

For these obvious reasons, this work remained in the margin, not well understood by art circles, nor well received by technolgists, both sides being captivated by the extremely rapid progress of mass market digital graphic tools end the emerging of a high end, high budget art-and technology paradigm, surrounded by a proliferating world of graphic design.

Digital technology was remaining absent from the traditional studio, more than anything because of the absence of learning environement adapted to its practices. Equipment and software remained , for most artists, prohibitively expensive, plagued by a rapid cycle of obsolescence and by the difficulty of absorbing the fundamental knowledge necessary to use this medium with the autonomy that is fundamental to art. From the technology point of view, gesture was becoming obsolete, the traditions of studio art were being marginalized into handycraft and personal therapy. Very large insitutions and corporations were dominating the public perception of where the frontiers of art were heading.

2. Nomadic workshops

By the end of the nineties, however new technologies for assembling computers, suface mounting, rising integration and miniaturization of components, huge increase in processing power made it clear that a very large number of machines would rapidly become obsolete, releasing an enormous quantity of "dead matter", boards, chips, components of all kinds that would contain, frozen in their dismemebered parts all the knowledge and skills needed to allow artists the possibility of revisiting and re-enact, with their own traditions, the fundamentals of digital tecnology.

Rather than limiting the focus of art and tecnology to high end, expensive, state of the art research institutions and projects, I was convinced that it would make sense to create low cost "interface" spaces between the traditional "gestuality" of studio art and the fundamentals of computer science and electrical engineering. Such interfaces could take advantage of the growing acumulation of electro-digital refuse and use it as a "raw" expressive medium, mix it with the skills and knowledge of the basic engineering education curriculum and dedicate themselves to the "staging" of these fundamentals as a form of art, thereby releasing from within the universities new, sustainable methodologies for technological outreach and renewal of investigative energies of young students

To seed and probe the feasability of such spaces i imagined a mobile art performance/installation workshop which would travel with compact kits of computing resources, parts etc... and propose to art communities, schools, galleries, museums, universities, a cycle of workshops introducing mixed groups of artists and students of engineering , art, arquitecture, comunication, social sciences to the fundamentals of the computer as a raw expressive media. The process would always begin with reviewing how to assemble a computer from obsolete or inexpensive off the shelf parts, invent its form factor, install operating systems, network machines together, connect them to simple devices and record the process, and all the questioning that it would raise. These recordings, notes, schematics, pictures, questions, answers, could be processed into a kind of documentary produced by way of an introduction to writing code with extreme economy of means. An art “with programming “, articulating the core vocabularies that should be shared by the many different groups of people confronted with computers as a common nexus of encounter

These "nomadic workshops" should keep away from emphasis on making or using "products" but rather focus upon production of symbolic value, explore an esthetics of basic electro-digital-computational functionality, as Art, and explore patterns of "gesture" proper to this new medium with a concern for radical economy of means and autonomy. Each workshop would result in some sort of recording (CD-rom) of a resulting installation/documentary on the fundamentals of electro-digital technology and its perception by the group or the community receiving the workshop.

The network of communities that had hosted these workshops could share an ongoing reflexion over the data produced by all the events of the cycle. A form of Collective intelligence that the open source software community had already actively begun to develop, but with a greater emphasis on integrating physical gesture, abstract, technical knowledge and expressivity.

Interestingly the explosion of internet and the virtual-technological euphoria of the last years of the XX century, deafened the ears of the academic world to this proposal: Techology had to be the ultimate. Research had to look far into the future, because the future seemed to be "starting up" every day. Economy of means was irrelevant, products and markets were going to solve the problems of the world. The credo of liberalization and open markets, however, was not matched by territorial traditions. Fierce, unpublished battles over intellectual property left no room for such a "free", purely educational endeavour.

By that time, i had lost my studio in Paris and resettled in Boston in 1998 as a visting scholar at MIT , and could confront these ideas with the state-of-the-art research on media that was going on at the insititute and begin to run experiments to concretize this project: A one semester course dubbed "building a computer/making Art" at the Massachusets college of art, a few workshops in various schools of New England. A continuous informal workshop process with numerous students of all types. Nevertheless i kept encountering unsurmountable difficulties at overcoming the resistence of established curricula and territorial interests.

Research in education was focused on marketing futuristic products totally dependent on expensive logistics and maintained an uneasy relationship with art. Studio artists, were deemed unreliable, overly narcissic, and largely confined to handcrafting and personal therapy. Resources and space for computing in art schools were being absorbed by the needs of graphic design and focused on learning how to use proprietary software packages and production tools. Nothing was being done to uncover to non-engineering students the nature of the underlying medium of electro digital technology, and the "culture of measurement" implied by this emerging world of "bits and atoms" and by the complex relationships in which their structures operate. A set of relationships that is nevertheless blatantly at the core of the ongoing mutation of our means of perception. This mutation is not only technological but also problematically cultural. Technology had come hard and fast, but its fundamentals had not been transmitted at the imaginary level to the vast majority of people in the world.

3. Studio of art and programming: a first design

By the year 2000 my efforts in Boston had come to a dead end. My quota of visa years was expiring, and no academic insertion to sustain this research was in sight. An unexpected internet correspondence with a young uruguayan musician had made me aware that the southern cone of latin america presented interesting characteristics for a third attempt. Good educational and cultural levels. Interest in the Arts. Substantial investments in information technology, yet deep economic crisis and scarcity of resources. Growing marginalisation of unemployed unscolarized youth. Disconnected public education networks.

A four months trip to Uruguay with a suitcase containing a kit of resources (power supplies, motherboards,one flat panel screen etc..) to run small scale workshops with grass-roots musicians eventually put me in contact with the director of the Institute of electrical engineering (IIE) of the Universitad de la Republica (UDELAR) in Montevideo. By april 2001 we had sucessfully ran a one month extension course as a joint project between the IIE and the school of fine arts and outlined the project of a curricular experiment to begin at the IIE in September 2001. This implied a profound change of the original project for the "nomadic workshop": migrate from a freer, artistic, intuitive process to a form of "sendentary" curricular culture and root it inside of engineering rather than in an art dedicated environement. Instead of bringing technology to the artist's studio we were going to bring the art studio methodology to the learning grounds of engineering.

The initial design of the course had to respond to basic necessities of EE students in the 2001 Uruguayan context: For the most part, these would come to the course with only a background of introductory programing in Pascal, a little of Matlab, and very little awareness of the rich possibilites of the many programing languages available. Another difficulty would be their reduced capacity for physical fabrication and integration. In many universities of the third world students spend most of their 4 first years of studies absorbing vast quantities of abstract notions, of theoretical knowledge, without any resources for a hands on, concrete, investigative application of these concepts, and with the perspective of an industrial vacuum in their local environement at the outset of their studies.

At the core, this course had to aim at diversifying their options to access programming languages and to apply software development to making simple physical devices run. Agilize and enrich them with respect to the cultural and social implications of the technology they were learning in the context of their country and of their region. Develop their fundamental autonomy with respect to the computer, taken holistically, in all its aspects, as a “raw” expressive medium. Manipulate, fabricate, assemble, install. Both at the level of software and hardware. All this at a negligible cost.

To meet the goals of the original project, we also had to create the conditions of an interaction of their discipline (EE) with students from other disciplines who could be interested in taking that course, like computer science, art or architecture, music etc... and as a perspective, explore the possibilities of them designing together a low cost mobile workshop that could project such an "interface" to schools and communities outside of the university. The incremental production of one form or another of "documentary" of the learning process had to be central in the articulation of the course.

A weekly “theory” meeting of two hours for all the participants would review some fundamental concepts of art, and of its current conditions in the world. Showing a profusion of images of reference works, proposing extracts of critical texts on the impact of technology over culture and society , articulating for a very diverse audience a relecture of electro-digital-computacional technology as a visual “landscape” to explore, imagine and reconnect at a fundamental level.

Starting from a pattern of direct questioning of the audience, each “teorico” would then expose in detail, and in a manner accessible to all, one technical topic immediately applicable in the lab: Basic control structures in programming. How to imagine the link between bare bits at the ground level of digital media all the way to programing languages and data objects. How to use timers in javascript. How to think about the architecture of a microprocessor. What is an oscillator. What is the difference between traditional procedural programming and object oriented programming. How to use a static ram-chip.

An essential point of the course was of its reference to "studio art". Throughout the 6 weekly hours of required presence in the lab, the point of view of Art allowed a displacement away from the usual "problems, products, solutions" modality that prevail in the competitive marketplace, and towards which enginering curricula are normally oriented, closer to the freer modalities of Art and "symbolic value" production: Use an internet browser to make web pages that are not meant to be read from close up for their usual "information content" but meant to impact as dynamic, architectonic, elements of visual art installations. Cutting up the pcb of bare desoldered motherborads and using the fragments to regenerate invented, modular, formal arrangements of funcional electronics. Combine simple network technology with stepper motor control to generate sound spaces. All things that would have no economic or cutting-edge value but whose visual, pedagogical, impact could be substantial. In the current situation of Uruguay, it could show first of all, that "something can be done", in a context of absolute scarcity of resources , to seed, among a much larger number of younger students, a collaborative culture of research and implication.

4. Developpement of the curriculum: TAP1

Although the text approved by the academic council classically outlined a detailed series of steps and skills that the course was supposed to cover, over the two weeks that it took for the 45 participants (among them 15 non engineers) to install the computers, connect them with old coax cables into a subnetwork (a PC with two network cards to do network adress translation from one IP of the insitute) it became rapidly obvious that such an program was completely inadapted to the mixture of students present in the course. Art and arquitecture students were eager to make wonders but would systematically stumble upon any question that they could not completley demystify according to their own rules, one answer leading to another question, etc.. The culture of algorithms and the dictature of syntax was completely foreign to them. Learning about dynamic libraries in a curricular, axiomatic way was totally out of reach. Electrical engineers had a embryonic, rigid, dry algorithmic view of programming coming from their earlier pascal-programming course, computer science students were fundamentally not so different, knew more about C and java but had very little capability for doing visually expressive things. The gap between open source environements and proprietary tools, the scant resources available on the students own computers made it impossible to have all of them dispose, with full autonomy, of similar programming tools to enter the course. It became clear that javascript enabled web browsers were the only "programming" environement that was readily available to everyone and that they could be used to teach or revisit the fundamentals of programming in an expressive context. Over the first two months of the course, we collected and explored a wide variety of javascript code snippets that students would constantly circulate among them, modify, revisit and gradually merged into a the seed for a presentation framework fitted with menus and a full interface structure. Arrays, control-structures, timers, mouse-events handlers, simple command interpreters began to merge into a flurry of small visually expressive projects involving images, text and sound. The more advanced students began to experiment with Java. Small "server-like" applets were embedded into webpages that could listen to ports and soon we had an external client fitted with rudimentary scripting capabilities sending dynamic web-pages across a wall of networked low-end machines running full screen browsers. The wall was displaying controllable sequences of web pages moving shapes made with blank formularies, images and text . Navigators could crash. reboot under control of small lower-level daemons written in C. The same structures were then used to control stepper motors from old 5inch diks floppy drives that were banging small steel rods on metal scructures and plastic bottles. A rudimentary world of bits and atoms which had little to do with state of the art technology but whose expressive impact was evident.

Next to this, some students had produced embryonic tutorials about installation of open source systems and the basics of java programming. At the end of the first year, the flat, calibrated curriculum necessary for approval of the course had been reinterpreted into something much more sustainable,wide open to diversity and the course had not collapsed. The work of the students had been essentially individual, although with a high rate of interchange between them. Final presentations were lively and led to animated discussions pointing to the same question: "why not more Art?". All this had been done with 40 students working in a dark, hastily improvised space of 4x4 meters, in the basement of the institute, separated from dusty storage bins by plastic sheets: making more "Art" in there that could be credible to the outside world's esthetics would be very difficult. Developping in there a more systematic, more sophisticated reappropriation of the electronics that could be mined from the large quantities of electro-digital refuse that we were accumulating was out of the question.

5. Making things. TAP2

The workspace used for TAP1 had been improvised in a corner of a disabled high-voltage laboratory built around a large scale, obsolete, disabled impulse machine. Retrofitting this 11m wide, 12m long, 14m high building into new labs involved a $100k budget that current economics of the times would postpone indefinitely. We thus proposed to the university to spend $5k to build a large steel platform around the top of the high voltage machine and use it as a much improved, and spectacular, workspace for the course's lab, arguing that the visibility of a successful, ultra-low-cost, art and technology program would eventually facilitate the arrival of a budget to retrofit the building. During the summer 2002 when the platform was eventually built, i was back in boston, working frantically in a basement lab at MIT to figure out how we could desolder efficently hundreds of rotten motherboards and circuitry of all kind.

At the onset of the TAP2, in the fall 2002, we had a bare, empty platform. No budget, no assistants. Over the summer, the loal currency had brutally lost half of its value, the country was losing 3000 skilled people every month and the University was on strike, yet there were more than 150 applicants to the course. One solid month of solitary labour (a lot of "inventive" carpentry and some crude electric wiring.) while the university kept striking, and a first sketch of the lab was up and running. The challenge was to introduce "making things" on a much larger scale. without any kind of budget for materials and pratically no tools. I had brought from Boston two efficient regulated 450degC heat guns and a fluke portable digital oscilloscope , installed in a corner of the lab a metal cutting guillotine and managed to have 4 soldering stations sent by mail from Belgium. The only material available to explore in a sistematic fashion would be the bare pcb from motherboards desoldered with heat guns to generate electronic components and solder. This pcb could cut in thin slices with the guillotine. We had a few protoboards to introduce a minimum of "clasical" electronics-learning methodology, and the challenge was to use the cut pcb to produce a scalable, sustainable accumulation of electronics projects.

Students were asked to build their computers, create a directory with their name into the disk of the NAT/server PC. They went on to revisit the body of javascript code inherited from the TAP1 and fabricate, with fragments of sliced pcb, a connector that could fit the floppy disk cable from an AT power supply into a protoboard or into any makeshift pcb structure they would make. Making this connector was a way to introduce all of them to simple LED circuits and to state the problem of producing modular components that could be interchanged between them and combined to make more complex electronics. The very size of the course began to generate more groupwork. Java native interfaces were use to control ports. The first dll's and shared libraries began to appear. A desoldered 8051 microprocessor was connected into a minimal "tight loop" and the frequencies of pulses from its adress pins were showing that is was working: they could test desoldered micros. Hard labour from a few students eventually fitted this processor with a monitor that could load assembled code through the chip's serial port. Microprocessors could now be used in the course. The TAP2 saw a profusion of inconclusive experiments of fabrication of clumsy modular structures that students would photograph with low cost web-cam digital cameras. The goal was to arrive at a crude Lego-like system of small electronic modules that could be made at zero-cost and completely replace the usual protoboards for developement of projects. By the end of the semester we had a first generation of home-made protoboards and made various types of physical structures, fitted with a wide variety of oscillators or PCports drivers flashing leds. Programming tutorials made by the students had grown into a method for collective learning. The presentation framework had grown into a sophisticated system of webpages generated automatically from a file-system tree by a java program. Incursions had been made into VRML and the culture of 3D graphics. About 90 students partcipated in the final presentations and approved the course.

6. Making art. TAP3: a method for massivity

In the fall of 2003 we had close to 200 applicants. Working with so many people required one more essential element: how to connect individual achievement to loosely differentiated group work, among people that had extremely different schedules, many of them working in outside jobs far from the university, and, more important, how to maintain the possibility a direct relationship between the teacher and every participant in the course at every phase of the process. To handle the flow we designed a web based formulary with a mysql/php driven database that required from each candidate to put in their data and to fill a note descrbing: why they wanted to take the course, what were their skills and know how, their interests, and what were the things whey wanted to do in the course. The php program generated a mailing list to sustain bulk exchange of mails between the teacher and participants.

Since the beginning of the cycle, students had been working from inside individual directories in one single hard drive, everything being accessible by everyone. A natural step was to create a mosaic-like web page displaying the pictures of every participant, with their name as a tooltip for the corresponding picture, each picture directly linked to their working directory and a personal web portal. It was now possible to interact at any time with individual progress of the student, in particular during the weekly “theory” meeting of all participants. Tight, structured groups centered about projects were abandoned and replaced by a few "research poles". Each pole had a directory with a file in which students attracted to the activities of the pole would add their name and e-mail.

Electronics pole. Information systems pole. Computational cinema pole. 3D pole. Programming languages pole. Disoriented students were encouraged to contact people in the poles of their choice and find leadership there. People with skills and projects were encouraged to seek help in concrete tasks from less autonomous students.

Our earlier model of home made protoboards was clumsy, difficult to make, but it had allowed to think about how to make simple modules, integrating one or a few chips and discrete components, that could plug in equally either into our home made protoboards or into commercial ones, and be interconnected with appropriately crafted flat ribbon cables. Copies of one type of module, like debouce buttons, oscillators, a static ram, byte hex viewers, bus adress decode module for an ISA card, stepper motor controllers, r2r resistor DAC’s , sound amplifiers, were being built and rebuilt by different people, undergoing mutations that made their making easier and cleaner. Eventually our custom protoboards mutated into much simpler structures, designed just to power and combine plug-in modules together. From there on, making self contained 14.3Mhz microprocessor "developpement" systems, with memory, connectors for ports, etc.. became feasible for the electronics "pole" as an incremental, modular vocabulary. A long way from the first contraptions of the begining.

The previous exploration of 3D tools had suggested that a very useful collective project would be the design of a virtual model of the lab itself, integrating the various scales, from the entire building down to electronics components. Achitecture students used Autocad to model the building and the lab, computer science students interfaced the MySQL database to the Blender3D open source modeller using its embedded python scripting capabilities and a Tcl/Tk GUI, and EE students created models for chips, resistors, condensers, pcb strips etc... A crude object model for complex structures was written in python to store the descriptions of the scenes as persistent text files re-using components stored in the database. A "virtual TAP" could now be explored in a web browser from a VRML description generated by Blender and we could use this model to design detailled proposals for customised atelier-labs to be projected elsewhere.

The programming language pole had centered about 2D graphics. Programs for free hand drawing that could store drawings as vector objects files. Using Java, and then using Python. Advanced students explored Haskell to generate computational 3d structures, outputting files that were compatible with the python readers we had designed for Blender. The work in the Haskell group eventually led to the fabrication of a Logo interpreter made with Java and sophisticated generators like Jlex. Using these techniques it was easy to add to the interpreter the capability to interpret snippets of this logo code, stored as records in a mysql database, as extensions of the language, generating fancy rythmic computational structures that could be fed in Blender.

The system pole had worked on the integration of linux into the lab for very low end systems, retooling popular "live-cd" distributions into "loopfile" systems that were easy to clone and could operate from within one file inside of a DOS hard-drive. A token ring installation of networked 386 linux machines with very little memory explored the esthetics of moving graphics across multiple screens.

Thanks to the unusual quality of their presentation, participants from the school of industrial design used their newly acquired agility with javascript to win a context for the design of a fluorscent tubes and mercury lamps recycling machine. Their interactions with electronics students had allowed them to think of a design that integrated control electronics at a level that industrial design students in Uruguay are no commonly exposed to.

By the end of the TAP3 we had produced the base for making, at extremely low cost, and from the contributions of many people, large scale visual art installations combining a vocabulary of modularized, sculpture-like electronics controlling a variety of visible events with their virtual 3D couterparts.

7. Conclusions

In many academic situations, earlier years of studies are spent mostly if not completely in absorbing a steady flow of theoretical, abstract, pre-formatted concepts with little accesss to hands-on practices of investigation. The creation of research environements using conventional, up-to-date technological norms is, in most cases prohibitively expensive.

By combining the expressive freedom of art and the rigorous demands of technological funcionality, TAP opens interesting avenues for bringing a larger number of younger students from a broader spectrum, to the experience and the "socialization" of research., fostering , at a sustainable cost, the seeds for what Manuel Castells calls " fabrics of invention".

According to their own comments on the course, engineering students, working together with students of other, more artistic or more communication oriented disciplines can benefit greatly from the mutual displacement of perspectve offered by a suitable reinterpretation of the tradition of studio art in the context of electro-digital-computacional technology. The use of the internet-navigator and its scripting capabilities provides an efficient way of introducing "expressive" programing to non-programmers and bringing them in working contact with tech-savvy students. Architecture and art students can rapidly make useful contributions to fabrication of physical structures for functional electronics, while demistifying progressively some fundamental aspects of the technologies involved in the projects. Young EE-students find in TAP an exposure to an operating system culture that is more common to computer-science students. Computer science students find, many of them for the first time, a physical contact with the basic processes of their medium that they might otherwise never get. The problematics of the social impact of technolgy can be shared by all.

Although the relative crudness of the devices fabricated with such "makeshift" means may seem derisory in front of the sophistication of current electro-digital-computational technlogy, they essentailly deal with the same fundamentals: bits, circuits, programing agility, modularity, abilty to focus on relevant detail in a very complex landscape and psychomotricites essential to research. Substiantial benefit to the students come from the fact that they must learn, in a freely expressive context, and with their own resources how to integrate, or simply become aware of, a large quantity of informations, languages, physical skills and critical points of view.

Massivity of enrollement which normally tends to weaken the efficacity of traditional courses, tends to become an asset in the TAP context, provided that a minimum of communication logistics can be maintained between the teacher and the students. This means: functional mailing lists, a mosaic web page of named individual pictures, a quick way to browse individual presentation notes and e-mailed comments associating them visually with their names and pictures, the simple, old fashion, physical device of a sign-in book for each of their presence in the lab. The initial requirement of building the computers they use, desolder and manipulate electronics refuse and generate their first documenting web pages by means of writing code, rapidly produces a sampling of the skills and the know-how in the group. The principal task of the teacher is then to identify the abilities of the most creative participants and bring their emerging work to the attention of many. The accent is then put on identifying among the students which are the ones that can take a degree of natural, consensual leadership in the developement of the various "poles". The loose structure of the work in "poles" seem to favour the diversification of a few thematics that emerge naturally and allow to impulse a large number of differentiated individual learning curves that can be monitored from the mosaic of the students web-pages.

Even though during the first weeks of the course a number of students appear disoriented by the complete freedom in the work, and, for some, by the apparent simplicity of the tasks, which contrasts with the sophistication of their theoretical courses, exposure to emerging realisations by the ones and the others with more dedication and concrete skills eventually pulls most of the students into a workflow. At two thirds of the course, we can observe a period of peak activity in the lab, as if the participants had finally begun to understand the nature of the process and it becomes quite obvious that everyone is growing and learning a lot of things. This in turn combined with the openeness and informal nature of the work seems to affect the most hesitant which have, in a number of cases, come to produce interesting work towards the end of the process.

Given Uruguay as the initial location of this experiment, an interesting and feasible development of this work could arise from the establishment and the interconecting of a number of similar courses and of their associated "taller-labs" in various universities of neighbouring countries in the southern cone of Latin America. Such a network would be particularily suited to explore a pattern of short-term, physical, academic interchanges for younger students (participating in these courses) that is significantly lacking in this region. The low-cost of the method and the freedom arising from the absence of questions of intellectual property (Artistic value of the work here is in "collective intelligence", just like it is now the norm in the open source software community) coupled with the ubiquitness and uniform features of discarded electro-digital refuse could allow the students to make immediate contributions to the spaces they visit, even for short periods.

The need for sustainable methods of re-connection of large social sectors to basic understanding of technology are immense in these regions. The costs of doing this are very high, and would depend critically of the existence of a generation of suitably trained and motivated "multipliers" apt at fostering circulations of knowledge of another type. A mixture of engineering and art/comunication students trained in the context of such a low cost network would provide an efficient training ground for such "multipliers"

Three years of experimenting with such a course reinforces my conviction that, in times of enormous increase in the complexity of our environements, we cannot afford the luxury of not including traditional Art in the diffusion and in the re-connection of the fundamentals of technology to the social fabric at large. Curricular spaces such as the one descibed here, operating from within schools of engineering, in collaboration with other parts of the university, would provide an interesting way of exploring the problem.