PIETRO GROSSI'S LIVE CODING. AN EARLY CASE OF COMPUTER MUSIC PERFORMANCE.

Giovanni Mori
University of Florence
giovanni.mori@unifi.it

ABSTRACT

Pietro Grossi has been one of the first pioneers in computer music in Italy. His work, however, is still quite underconsidered because his art's concepts was judged utopistic, without a connection with contemporary cultural manifestations and harshly anti-academic. Instead, in my opinion, it seems to be now the right moment to revalue his work, in order to understand from where some computer music practices have their roots. In this article, I compare the concepts and the work methods developed by Grossi to those employed by live coders. My aim is to demonstrate that the Italian composer's concepts were not only ahead of his time but also anticipatory of future developments.

1. INTRODUCTION

Since my first live coding experience, I feel very enthusiast about this innovative technique. I have immediately perceived something different in this way of making computer music, compared to other similar types of performances. Here, liveness perception is very strong, audience is completely aware to attend to a live event. Listeners, instead, do not generally experience the same sensation, during other types of computer music concerts and this creates a sort of alienation and discomfort among audience members.

I have attended to my first live coding symposium at the University of Sussex in Brighton. Speaking with people and listening to the paper, I became increasingly aware that live coding appears to be very similar to the experiments made by the Italian computer music pioneer Pietro Grossi, who has done his first experiences in algorithmic composition about 50 years ago. After a review of my MA thesis, which was exactly pinpointed on Grossi's computer music and programs, I am feeling quite sure about this familiarity. Naturally, there are some differences between Italian composer's and his younger "colleagues'" approaches, due to the diversity of the technology employed. This article's aim is to help the rediscovering of Pietro Grossi's work and theory, and to bring his thoughts to the attention of those I consider in some ways his successors. I also believe that Italian composer's work may have had an influence, perhaps unconscious and indirect, on the formation of the live coding music practice.

2. INTRODUCTION TO PIETRO GROSSI'S COMPUTER MUSIC

Pietro Grossi has been an "underground" figure in the Italian and European music field during the second half of the XX century. He was born in 1917 in Venice, but he moved very young to Florence, where he became the first cellist in the orchestra of the Maggio Musicale Fiorentino, a well-known Italian music festival based in Florence, during the 1930s. At the beginning of the 1960s, he become interested in analogic electronic music and he was guested at the Studio di Fonologia Musicale in Milan for fifteen days. This studio was famous because there, Luciano Berio, Bruno Maderna and some other famous music composers were working. The result was *Progetto 2-3*. Some algorithms designed the sounds' architecture by intersecting a group of six sine waves whose mutual ratio was in slow evolution. After this experience in Milan, he was so enthusiast about this new music technology that he decided to found his own private phonology studio called *Studio di Fonologia Musicale di Firenze* or S2FM. This was one of the first private owned studios in Europe. Moreover, Grossi soon developed a large network of contacts with many important electronic music studios around the world.

Since the late 1950s, big mainframe computers became to spread throughout Western countries and, at the same time, many technicians and musicians started to experiment these machines' music possibilities. Among pioneers, we can surely list Leonard Isaacsson, Lejaren Hiller, Gottfried Michael Koenig and, last but not least, Max Mathews. Grossi began to be interested in computer music during the first half of the 1960s, when he hosted a radio program centred around "innovative music" in general (Giomi1999). However, the first Grossi's experience with calculator took place in Milan, in the Olivetti-General Electric Research centre. Here, aided by some internal technicians and engineers, he managed to compose and record some of his first computer music works. They were, for the most part, transcriptions of Western classical music. However, there were some exceptions, for example a track called Mixed Paganini. The name was a consequence of the original music material, which comes from the Fifth Paganini's *Capriccio* for solo violin. The Olivetti GE-115 computer played that piece in various ways: inverted, accelerated, retrograded etc. Practically, Grossi modified, aided by some rudimental music programs, the original sound material. People that was able to listen to Grossi's music complained about that because they felt this approach as a "profanation" of the art's sacredness and "aura". A later collection of Paganini's *Capricci*, recorded in Pisa, was reviewed by Barry Truax on Computer Music Journal (Truax1984).

In this early period, Grossi was not very satisfied by his work at the GE-115 computer because he cannot experiment freely. Moreover, the corporation's steering committee has lost much of the previous interest in the Venetian composer's music experiments. Then, Grossi began to look for a different place for developing his thoughts, and he finally managed to be admitted at the IBM Research Centre in Pisa, inside the CNR Institute (*Centro Nazionale per la Ricerca*: National Research Committee). Here he was able to work with a video terminal and alphanumeric keyboard, a very new technology for that time. Thanks to this device, he started to learn FORTRAN programming language and developed his first music software DCMP (Digital Computer Music Program). This was one of the first live coding programs in my opinion, because, using that, the performer was able to produce and reproduce music in real time by typing some specific commands and the desired composition's parameters. The sound result came out immediately after the operator's decision, without any delay caused by calculations. There were many reproduction choices inscribed in this software: it was possible to save on the computer memory pieces of pre-existing music, to elaborate any sound material in the hard disk, to manage the music archive and to start an automated music composition process based on algorithms that worked with "pseudo-casual" procedures. There were also plenty of choices for piece structure modifications. One of the most important aspects of Grossi's work was that all the interventions were instantaneous: the operator had not to wait for the computer to finish all the requested operations and then hear the results. Data calculation and sound reproduction were simultaneous. This simultaneity was not common in the computer music field of that time and Grossi deliberately chose to work in this way, losing much on the sound quality's side. His will was to listen to the sound result immediately. He was a musician and he probably could not stand to fill a chart or to write a long string of commands and parameters and then wait the whole calculation and recording process before being ready to play the result. He wanted "all and now", as he said very often (Giomi1999). With that expression, Grossi would address that when he inserted some commands on the computer screen, he did not want to wait anything before hearing the result. He defined himself as "lazy", *pigro*[1] in Italian. Lazy address, in Grossi's sense, to a person who is aware that his or her time is limited and do not want to waste time in doing useless things or in waiting for something when it is not necessary. His or her actions should be as efficient as possible, to save time for enjoy the life fully. Eric Raymond[2] has stated some similar for describing the hacker's philosophy. This approach should sound familiar to live coders as well, considering that they are musician but also programmer and many of them seems to have something in common with free software community, in Stallman's sense (Stallman2010).

The DCMP was compiled in the early phase of computer technology development. At that time, the calculation resources were low and, to obtain the just cited real time reproduction, it had to ask for very low quantity of data. Therefore, the Venetian musician chose to write very light software, able to modify only parameters that required a few calculation resources: pitch and duration. Timbre synthesis needed a big amount of data, so that choice was temporarily discarded and all the sounds were reproduced with square wave timbre. This waveform was generated by extracting the binary status of a motherboard's exit pin controlled by the software. This exit had only one bit, so the sound wave generated was the result of this bit

[1] Grossi as nickname in some of his records used *Pigro,* or better *PiGro*. It is a play on words made by using his name first letters: Pietro Grossi.

[2] Eric Raymond's post on Linux Today: http://www.linuxtoday.com/developer/1999062802310NWSM

status changing. In this way, the computer did not employ any resources for calculating the sound synthesis, saving them for music production process. Grossi was not very interested in the quality of sound output in this first phase in Pisa. What he cared particularly was to be able to work in real time, or, in other words, to have the choice to listen immediately to what he typed on the video terminal's keyboard (Giomi1995).

Some technology improvements permitted Grossi to implement timbre elaboration around the first half of the 1970s. In this period, the CNR steering committee decided to project and build a brand new and innovative sound synthesiser, digitally controlled but with analog sound synthesis technology. It was launched in 1975 and called TAU2 (*Terminale Audio 2a versione* – Audio Terminal 2nd version).

Grossi was the new machine's software developer. For making this, he took as a reference the previous DCMP, adding to it many new features and a brand new section dedicated to TAU2's management. The new program's name was TAUMUS. The new software and hardware system could play up to twelve different voices simultaneously. These twelve voices were divided in three groups, composed of four channels each. The operator could choose to assign a different timbre to every single group, which was modulated using additive synthesis with seven overtones. Every overtone could be controlled individually by software.

It may be interesting to mention that using these two music systems - DCMP and TAU2-TAUMUS - Grossi developed his most innovative artistic conceptions as those of, for example, infinite music and automated composition. With the first, Grossi addressed to a music production process without predetermined end: a music that flows forever, without interruptions. With automated composition, instead, he defined a music created without any human intervention after the start of elaboration process. Every decision was a computer's task, or better, a software's task. This last one is an extreme example of aided music composition. This fundamentalist approach may have stimulated the developing of automated process inside computer music practice. To make an example in the live coding field, if someone types on software's main window a command to let the computer reproduce a beat pattern, the computer will play it automatically, and the performer can dedicate his or her attention to the next task. That pattern will not stop without external intervention. Andrea Valle[3] states that these two approaches own to "instrument logic" and "machine logic". The instrument needs continuous stimulation to be operative. The machine, instead, need only to be programmed and then it will work respecting that program.

In the first period of music experiments with DCMP, the calculation resources available to Grossi were very low and this did not enable the musician to change the result in real time while the process was running. Then, it was necessary, if he wished to modify one or more parameters during the elaboration, to stop the entire process, make the desired changes and restart it from the beginning. When the process was an automated composition, it was not possible to vary any parameter at all, because the program took every decision autonomously. But, as it will be shown later, there are some evidences in Grossi's music applications, especially the latest ones, that shows his aim to develop the sound output in real time, in a surprisingly similar way to that of live coding performances.

3. GROSSI'S SOFTWARE AND HARDWARE

First, I would like to introduce the DCMP's architecture and functions. Grossi started to develop his program in 1969, when he began his career at the CNR in Pisa. The software was written in FORTRAN and designed to work on some different machines: IBM 7090, IBM 360 model 30, 44 and 67 (Grossi and Sommi 1974). DCMP was provided by many different commands, both musical and non-musical. The last ones were principally commands dedicated to the musical archive management: erase, move, rename files and so on. The first ones, on the other hand, were designed to modify pre-existing pieces or to compose and play new music by using both automated and non-automated compositional procedures. The system could play about 250 notes between the frequencies interval 27 Hz - 4900 Hz. The minimum step between two sounds was a third of a semitone. The whole pitch range, then, was approximately that of a grand piano. It was possible to play notes addressing to them in various ways: by using alphabetical or Italian system or by using the note's number in a default note chart. It was not possible to manage neither the timbre nor the dynamic nor, finally, the number of voices, because of sound synthesis method, as explained above (Grossi and Sommi1974).

All the music pieces created with the DCMP could be saved in the computer's archive. From there, the operator could recall any track with specific commands, upload the chosen one in the work memory and

[3] See Andrea Valle's interview: http://www.soundesign.info/interviste/intervista-con-andrea-valle

elaborate, modify and finally play it. The sound processing instructions were numerous: MIX, for example, mixed two tracks' parameters pre-uploaded in the working memory. It can exchange the note's durations of one track with those of the other, or sequencing two notes of the first and one of the second track, and so on. Some other interesting commands were: GOBACK, that reproduced the track placed in the work memory from the end; INVERT, that inverted the pitches' ratio in respect of a pre-determined value; SHUFFLE, that mixed randomly two tracks' parameters; and so on. The operator could stop the music reproduction in every moment, change any parameter, and restart the process. However, it is not possible to define this as performance, because the system needs to stop the elaboration process to act every kind of intervention. It were not interactive (Grossi and Sommi 1974).

There were another choice for creating music employing DCMP: automated composition. There were some commands dedicated to this function, but the most interesting one was, in my opinion, CREATE. The only action requested to reproduce music through this command was to type the desired total piece duration on the screen and start the process. The program then started to play random notes generated by specifically designed algorithms based on Markov and Lehmer chains. Those algorithms were studied to create as less repetitive flux of sounds as possible. The time limit for automated compositions were fixed by software at about 300 years, then almost endless for a human being (Grossi and Sommi 1974).

Thanks to the technology advancements, in 1972 Pietro Grossi and the CNR team began to project the TAU2 synthesiser and the TAUMUS software. The TAU2-TAUMUS system was able to reproduce about the same note range of DCMP system. There were many differences, however, between them. The most important one was that, by using the newest one, the operator could play many melodic lines with different timbres. In fact, the TAU2 was able to reproduce up to twelve voices simultaneously, grouped in three different set. Each set could assume a distinctive timbre. As I have already written, the sound synthesis technique employed was the additive synthesis, with seven overtones (Grossi and Bolognesi 1979).

Another crucial TAU2-TAUMUS' improvement in respect of DCMP was the *modelli modulanti* (modulating models): they were a sort of patches that acted on some musical parameter. The most interesting aspect is that these patches worked in real time, and the operator did not need to stop the process to change the sound output, as happened before. This was a critical innovation under the performative point of view, because then Grossi was able to play and to interact in real time with the software, by typing instructions on the keyboard without stopping the sound flux. It is clear, then, that his interest was to be able to work with sounds with a real-time interaction. In this way, the computer becomes a true music instrument and not only a sort of augmented music player.

All these aspects get Grossi very close to the newer live coding practice. Moreover, what I have explained shed more light on the Venetian's research direction, making us wondering about his potential goal if he was remained in a stimulant research milieu. In fact, in 1982 the TAU-TAUMUS system developing was abandoned for the increasing number of bugs and failures. After this long period in Pisa, he began to work at another CNR institute in Florence, called IROE, where he continues his experiments with a new synthesiser: the IRMUS. This last one, unfortunately, was very less powerful and ambitious than the TAU2, because it could reproduce only two voices simultaneously. Nevertheless, IRMUS was very flexible on the timbre side, thanks to the FM synthesis method adopted. Unfortunately, this synthesiser was not sufficiently powerful and stimulant for Grossi and he decided, after a brief period, to quit his music experiments (Giomi 1999).

Grossi, however, was a very curious person, so he could not stay without a computer and then he decided to buy one of the first popular PC on the market, the Commodore 64, and to make experiments in the computer graphics field. He came up to this decision after discovering that these machines were more powerful on visual than on audio processing side. Therefore, he started to adapt all of his programs to the new project (image elaborations in real time), recycling also his concepts about sound processing (Giomi1999). Even on this aspect, there may be a link between Grossi and live coding. In fact, in Lisbon, during a performance organised for the conference on Live Interfaces, I have seen that live coding is not only employed in music but also in image processing practices. Then, probably, further research on this aspect will underline many other affinities between Grossi and live coding movement.

I would like to return briefly to the TAU2-TAUMUS system, because at the end of the 1970s it faced an important improvement: it became possible to control the synthesiser remotely. This new feature was deemed so important that the software name becomes TELETAU. Then, it was possible to generate sounds and listen to the result immediately, ideally everywhere, thanks to a telephone connection between the

TAU2, the computer and a remote terminal (Grossi and Nencini 1985). However, similar experiences were possible also with DCMP and the first one took place as early as 1970 and was called “musical telematics”. Grossi made his first experience of this kind during a conference on technology in Rimini in 1970, where the musician reproduced many of his compositions and random sounds as well, by employing a video terminal connected via telephone to the CNR's computer in Pisa. RAI, the Italian public broadcasting company, lent its powerful FM radio bridges to send back sound signals from Pisa to Rimini. It is likely to be the first official experiment of musical telematics in the world (Giomi1999).

After this exciting event, Grossi went on developing his project along the others and the result was the just cited TAUMUS' update: TELETAU. In a similar way to that of the DCMP, this software enabled whoever connected to the BITNET computer network, to exploit remotely the CNR computer calculation resources and to immediately listen the sound results produced by TAU2. Practically, every person connected to this network was able to use the TAU2-TAUMUS system as if he or she was in Pisa in front of the machine (Bertini et al.1986).

Unfortunately, this service lasted only for a few years. There were many reasons for this decadence: the rapid ageing of the TAU2 synthesiser, with frequent breakdowns and bugs that increased dramatically the maintenance effort and costs; the rapid diffusion of personal computers; the high cost of data transmission and, last but not least, the low quality of sound output due to slow data connection.

This working practice was strongly promoted by Grossi because, since his career's inception as electronic musician, he struggled to develop a communitarian and egalitarian work method, based on shared knowledge, on free re-elaboration of pieces created by different musicians and on spreading as much as possible this conception to other similar working groups or single artists. In addition, this approach appears to me as anticipatory of that typical of live coding milieu. In fact, in my view, live coders too undertake to create an egalitarian community, based on almost the same philosophy promoted by Grossi. Nearly all the programs used by live coders are open source, or free software in the Stallman’s sense (Stallman2010), and people inside this community often modify and create customized software by reprocessing those of others. Moreover, I have assisted in Birmingham to an online performance at the Network Music Festival, where three live coders located somewhere around the world (Alex McLean, David Ogborn and Eldad Tsabary) met in a virtual place (the remote server) to play the concert we assisted to. I think that Grossi's work on musical telematics pointed out to similar kinds of music making.

Probably however, his artistic conceptions were too ahead of his time. The technology and probably the culture were not still ready to accept the detachment of music from human agency. The process of body dematerialisation has started long time before Grossi's work: radio and telephone inventions gave birth to all the subsequent technology transformations. Electricity carried out a central role in this process (McLuhan1964). Probably the development of mediatization, whose implication is described very well by Philip Auslander in his well-known book *Liveness*, is a natural developing of the mass media technology and requires a recontextualisation of the body's role in our contemporary society. The physical co-presence is no more necessary to communicate. It is apparent that, with the spreading of social networking and the Internet in general, the most part of communication between humans, at least in Western societies, passes through mass media. Live coding stands inside this process as protagonist, because its practitioners seems not to be astonished in front of a remote driven performance. Grossi was probably more tied to the performer’s materiality, because he never plays computer for a remote public in an acousmatic setting. He dematerialised the music instrument, but not the musician. Live coders instead do not have problem to express themselves through online connections. Nevertheless, we all express ourselves regularly through the same medium when we use social networks or the Internet in general and our identity is increasingly constructed through the Internet.

Finally yet importantly, Grossi took advantage of the TELETAU system by teaching to his pupils in Florence (actually he has taught at the Conservatory of Florence since the early ‘60s) without moving from his class. He was also the promoter of the first Electronic music course in an Italian conservatory in 1962 and, as well, the first Computer Music course in 1970s.

4. CONCLUSION

In this article, I tried to show the main characteristics of Pietro Grossi's computer music experimentations. Beside this, I have underlined the most striking similarities between the Italian algorithmic music pioneer and the live coders' works, in particular apropos of the respective processes of music making.

After having synthetically explained Grossi's approach, I would like to conclude this presentation with some reflections on the two central experiences illustrated in this article, to give more weight to these affinities.

I have said above that Grossi began to learn programming immediately after his arrival in Pisa, with a well equipped studio and a numerous team of engineers. He started studying programming languages because he knew precisely what he wanted and he was very focused on realising them. He strongly believed that teamwork was the better way to obtain a good and satisfying result, so he struggled to become an active member of this group. This is a strikingly similarity between Grossi and live coders, because also many of them have developed their own music software, sometimes aided by a community of developers, placing themselves on the edge between the world of musicians and that of programmers.

Another common feature between the Venetian composer and his "heirs" is that they all demonstrate a strong interest in sharing their knowledge on egalitarian basis, by using or writing programs without copyright licenses and by creating a multidisciplinary and collaborative work community. In fact, Grossi was one of the first artists that believed and worked in a way typical of what today is defined open source or free software field. He had very soon adopted a working method that anticipates the principles of the free software communities. To confirm this, it is sufficient to say that, starting from the work at S2FM (Music Phonology Studio of Florence), he gave a collective name to the records created there name because he thought that they were products of the whole community's work. Moreover, the whole studio staff often recycles pieces recorded by external artists to create something else, in a sort of proto-remixing practice. Grossi also gave his audio materials freely to musician who requested them, but also by sending tapes all around the world. Inside every parcel, he always inserted a little letter to invite the addressee to rework the tape content personally, without any limit and without asking any copyrights. Something similar happened as well for Grossi's software in a later period, when he was able to spread his programs out of the CNR studio in Pisa. In fact, the TELETAU system was explicitly created to broaden TAUMUS-TAU2 employment.

When Grossi began to work at home with his new Commodore 64, he went on following the previous path. He was in constant contact with many artists and, as soon as he managed to own an Internet connection, he designed his own website and shared all his new and old software, inviting visitors to download and modify it freely. Thanks to this pioneering behaviour, he has been recognised as one of the open source music software's ancestors among some hacker artists (for example: http://www.hackerart.org/corsi/aba02/taddepera/index.htm).

Another convergence point between Grossi's and live coders' approach is the TELETAU system: the use of a network to make music from remote places. Grossi was able to program a system that, by using a video terminal, it was possible to run the music software in Pisa remotely and to hear immediately the sound result. Live coders use sometimes a similar approach. The above-cited example I made about Network Music Festival in Birmingham (UK) suits perfectly here, but there are also some other examples as that of BEER ensemble, which work and play through a VPN connection to synchronise and to send music to each other (Wilson et al. 2014). In many live coding contexts, then, the network is a crucial tool for working in real time. In my opinion, Grossi was experimenting precisely in this direction. As I have hinted above, however, the live coder's case sometimes imply that the performer becomes invisible, located in a remote place in respect of that of the sound reproduction. Grossi instead wanted to be present in the same place where the audience heard the sound.

Another important similarity between Grossi and live coding is the user interface. In both cases, the music tool employed by the musicians is the text, and both require written commands and parameters to obtain precise audio results. Additionally, each refers more to the programmer's perspective than to the musician's one because there is not a direct correspondence between gestures and the sound heard (see note 3). In Grossi's case, however, choices are fewer than in the live coding one. Nonetheless, the performing practices of each example seems to be quite similar, because in every case, there are commands that recall precise algorithms, launched and modified normally by typing their name and parameters on the screen. Consequently, it seems that keyboard, screen and text itself have together a central role in playing and representing the musical ideas.

Finally, it may be important to cite briefly the *modelli modulanti*'s case, the above-cited patches that modify the final audio output in real time. The Grossi's effort to develop these little programs demonstrates, in my opinion, his intention to interact with the computer on the fly. By consequence, this action switches Grossi's musical practice from music composition to performative context. This crucial aspect of the Grossi's computer music work is worth to be emphasised here, because it carries the Venetian composer very close to live coders. The *modelli modulanti* demonstrate Grossi's strong interest to employ the computer as a real music instrument, or, in other words, as a device able to translate music thoughts in sounds in a performative way. Grossi himself confirm his interest about real-time music procedure, speaking about this aspect very often in his books and articles (Giomi1999).

To conclude, then, after having explained all these similarities between Grossi's and live coders' work (a predominant textual interface, experimentation of online performances, an egalitarian and communitarian approach to programming and to culture in general) it may be appropriate to define Grossi as a proto-live coder.

REFERENCES

Auslander, Philip, 2008. *Liveness: Performance in a Mediatized Culture*, New York, Routledge.

Bertini, Graziano et al. 1986. “TELETAU. A Computer Music Permanent Service”, *Proceedings of the International Computer Music Conference*, San Francisco.

Calliau, Robest, Gillies, James 2000. *How the Web Was Born: The Story of the World Wide Web*, San Val Inc.

Collins, Nick et al. 2003. "Live Coding Techniques for Laptop Performance", *Organised Sound* 8(3): 321-330

Giomi, Francesco, Ligabue, Marco 1999. *L’istante zero. Conversazioni e riflessioni con Pietro Grossi*, Firenze, Sismel Edizioni del Galluzzo.

Grossi, Pietro et al 1979. *Modalità operative del Taumus, software di gestione del Terminale Audio TAU2 2° versione*, Pisa, Internal CNR report.

Grossi, Pietro, Nencini, Graziano 1985. *Teletau. Software sperimentale per la Telematica Musicale. Release 0.0 – Luglio 1985*, Pisa, Internal CNR report.

Grossi, Pietro, Sommi, Giorgio 1974. *DCMP versione per il sistema 360/67*, Pisa. Internal CNR report.

Magnusson, Thor 2014. “Herding Cats: Observing Live Coding in the Wild”, *Computer Music Journal* 38(5): 8-16.

McLuhan, Marshall 2011. *The Gutenberg Galaxy. The Making of Typographic Man*, Toronto, University of Toronto Press.

McLuhan, Marshall 2001. *Understanding Media*, London and New York, Routledge.

Stallman, Richard 2010. *Free Software, Free Society: Selected Essays of Richard M. Stallman*, Boston, GNU Press.

Truax, Barry 1984, “Review of Pietro Grossi: 24 Capricci by Niccolò Paganini”, *Computer Music Journal* 8(1): 59-60.

Wilson, Scott et al. 2014. “Free as in BEER: Some Explorations into Structured Improvvisation Using Networked Live-Coding Systems”, *Computer Music Journal* 38(5): 54-64.