Denis Lorrain

Computer Music Does Not Exist: Only Music

When a finger points to the moon, the imbecile looks at the finger.

Chinese proverb

I have started computer music in times when it was really something in itself and, of course, relatively new. After the pioneer years, mind you, but still newish and exotic.

The computer was then mostly considered as a purely scientific number cruncher or, at best, in a more familiar manner, as a practical accounting machine. Musicians and laymen did not even figure how this calculating automaton could be involved with sound — not in the least with music. The very fact of relating computer to music seemed incredible, improbable. The consequence of this discrepancy, at least for beginners, was to promote computer music as a sort of exciting technological feat in itself.

The challenge was indeed great. Together with computer graphics, sound synthesis constituted the first potentially artistic applications of informatics which imposed serious technological constraints on the speed and power, random access memory size and storage capacity of computers. Experiments in the field of automatic generation of text and poetry were then also carried on, but with much milder constraints. I remember a famous professor and composer, an authority in analog electronic music, who deemed sound synthesis as impossible, or at least impractical, stating that it necessitated the computation of *so many thousand numbers per second* of sound! In this the estimated professor was somewhat right: my first experiments with sound synthesis at the university of Montreal were laborious! But nothing is impossible in the eyes of a young enthusiast...

This article is not intended to be a detailed autobiography. Nevertheless, I take the opportunity, from where I am now standing, to briefly look back at my path, and would like to mention the prominent people who have played a role in its meanders.

My involvement with computers started in 1970 with some friends[1] during my music studies. We immodestly called ourselves the *Groupe d'informatique musicale*. We started by doing what we could: studying Jean-Claude Risset's *Catalogue,* getting us musicians initiated to programming in FORTRAN and to some mathematics, efforts and unending discussions over Xenakis's book *Musiques formelles* in the original french edition[2], etc. But we also aimed at concrete sound results! At that time, the university of Montreal music faculty did not even offer an analog recording or electronic music studio. In this, I am rather exceptional for my generation, having experienced digital sound synthesis *before* accessing an analog studio later at McGill university[3].

With the support of the university[4], we managed to organize a visit to Max V. Mathews at the Bell Laboratories in Murray Hill, New Jersey, near New York. I will never be grateful enough to the memory of this generous man. He spent one whole day of his precious time with us three young visiting neophytes. At that time he was developing real time environments — GROOVE hybrid system, *Conductor* program and control interfaces. We also met Emmanuel Ghent and F. Richard Moore, who was developing innovative hardware digital oscillator boards. This memorable day concretely resulted in our return trip with kilos of punched computer cards: the FORTRAN code for *Music V*.

We implemented Music V on the CDC[5] mainframes of the university Computing Center. A CDC-1700 first read the deck of cards for *Pass 1* and the actual score, writing its output on digital magnetic tape. *Pass 1* performed a pre-processing of the score, quite equivalent to what modern environments

---

[1] Robert Dupuy initiated the group. He was a computer science graduate and worked as programmer in the university Computing Center. Alain Fortin was a fellow music student. Robert Léonard and, to a lesser extent, Jean-Marie Cloutier, both professors, also participated.

[2] Later expanded and translated as *Formalized Music*.

[3] The McGill Electronic Music Studio, headed by Paul Pedersen, mostly comprised prototypes built by Hugh Le Caine, a relatively large Moog synthesizer, and tape recorders.

[4] Thanks to Jean Baudot, Director of the Computing Center, and Jean Papineau-Couture, Dean of the Faculty of Music.

[5] Control Data Corporation, rival of International Business Machines (IBM).

would realize with macros. The deck of cards for *Pass 2* was then read, which processed the output from *Pass 1* and wrote on another magnetic tape the data ready for *Pass 3*, which was in charge of computing the actual sound samples. This ultimate *Pass* was batch processed on the CDC-6600, one of the largest mainframe time sharing computers then available. It occupied a dedicated area, more than one hundred air conditioned square meters of special flooring, tended by technicians forming a sort of clergy serving a deity's temple: we handed out the magnetic tape resulting from *Pass 2* with the *Pass 3* deck of cards over a counter and... just went home!

After a certain time, ranging from a few hours to a couple of days, we collected a final magnetic tape containing the actual sound samples. Severe technical constraints limited us to 8 bit samples at sampling rates of 8000 or 16000 per second — i.e. theoretically 4 or 8 kHz Nyquist frequency with 48 dB of signal to noise ratio! Then followed a long errancy through deserted corridors of the soviet style university main building... We reached a DEC[6] PDP-11 computer belonging to a neurobiology research team. This computer ran a dedicated utility program, reading the samples from our *Pass 3* tape, and sending them to a digital to analog converter connected to the *y*-axis input of a cathode ray tube display (CRT), from which we ultimately derived the resulting sound signal into a portable analog audio tape recorder. This could happen only at night, in fan noises and the smell of formol, because the computer was used in the daytime for other purposes, typically experimenting on pitiful trepanned rhesus monkeys and monitoring their brain activity on the CRT display through the same *y*-axis digital to analog converter.

You may figure that this tedious process did not allow for any approximate trials in preparing the Music V score data. The benefit of such travails is the instillation of the careful and thorough preparation of all code and data, which definitively characterizes one's work style.

These sound synthesis adventures certainly were the most time and effort consuming. Nevertheless, I simultaneously continued my musical studies and got involved in more abstract, so called symbolic computing, which later became my main center of interests. As a team, we ambitiously attempted programming an environment for musical counterpoint, including an interactive graphic interface using a CRT and *light pen* — more than ten years before the appearance of a device called *mouse*. This revealed itself a crucial landmark in my conviction of the profound difficulty of algorithmic emulations of musical tasks. Strongly attracted by Xenakis, we visited him

[6] Digital Equipment Corporation, mostly specializing in smaller laboratory computers.

during a few weeks in Bloomington, Indiana, again with university support. We studied his mathematical approaches, especially probability theory (*stochastics*) in my case, and programmed compositional algorithms.

The U.S. certainly were in those years the most advanced in computer music, and particularly in sound synthesis. However, the appeal of Europe had always been so strong on my wife and myself that we decided to migrate to the old continent. A few prominent contacts helped me initiate this new stage and led to all further developments.

I started with a first year in the *Intituut voor Sonologie* in Utrecht, Netherlands, before settling in France. There I renewed with Xenakis and the CEMAMu team in Paris. Introduced by Emmanuel Ghent, I had written to Jean-Claude Risset and he very openly proposed to meet me in a Boulevard St-Michel café. I owe my career to this most generous and righteous man. He still honors me with his friendship. He later took me as visiting composer in Marseille-Luminy, and subsequently, with Michel Decoust, and later David Wessel, hired me at IRCAM. In this institute, I had the chance to participate in many fruitful collaborations with, among others, Gilbert Amy, Philippe Manoury and Marco Stroppa, who shared the best part of my twenty years as professor in the Lyons *Conservatoire national supérieur de musique*. Jean-Claude Risset also associated me as consultant in the preparatory phase for the Karlsruhe ZKM, which brought the opportunity for me to meet Thomas Troge, who later greeted me with utmost kindness and openness as professor in his *Institut für Musikwissenschaft und Musikinformatik*.

Upon arriving in Europe, I also visited Pietro Grossi in Florence and Pisa. Pioneer of computer music in Italy, he practiced an approach at once naive and very radical — somewhat similar to that of Max Mathews. He had implemented self-developed sound synthesis programs at the CNUCE computing center in Pisa.

I also wish to evoke Bruce Mather, André Riotte, John Chowning and Michel Guiomard, with whom I have not had regular or very personal contacts, but for whom, as teachers or connections, I experienced warm and rather filial feelings.

To all these great professionals, and some of them friends, I am deeply grateful.

During the longest part of my activities, I have kept a parallel interest in both sound synthesis and signal processing, on one hand, and symbolic computing on the other. At the Lyons Conservatory, I have mostly taught in both domains equally. But I must say that my approach of the first field was rather simplified and focused on concrete musical applications: a sort of "Signal processing made easy for musicians by a musician".

In the course of time, however, I have gradually turned my preference towards formalism and symbolic computing, and concretely allotted more and more time to my lectures about LISP and an increasing collection of mostly self made examples of musical applications implemented in this language. Parallel to this growing importance of formalism and symbolic programming in my activities, I can retrospectively attribute my decreasing involvement with sound synthesis, sound transformations and spatialization, and signal processing generally, to the orientation taken by my activities at IRCAM during their last ten years.

Indeed, since the early 1990's the Institute founded all, or nearly all, its musical productions, and some of its research and development, on Max/MSP. I do not want to seem definitive or too critical about this wonderful environment, and do not need to insist on all its positive features. There are probably very personal causes to my gradual technical disinterest in the production of music within this environment — although I must add I have never been a genuine Max/MSP expert, such as I have known at IRCAM[7] for instance, and that I have ceased to follow closely its evolution.

The first reason is the lack of an explicit and readable structure of (sub-) patchers within Max/MSP patches, which a first-sight "reader" discovers as a discouraging collection of seemingly unrelated or clone windows scattered all over the screen. This must be compensated for by a very special and powerful mental ability which I probably miss. The second and minor reason is the multiplication of, and increasing dependency on special purpose objects and externals, sometimes of doubtful origin or poorly documented, interfaces of interfaces etc., which gave my sclerosed mind an impression — or fear — of loss of control; and unfortunately, in virtue of the rigorous apprenticeship described above, I remain a man of control... A third delicate point about Max/MSP is however not the least personal: it is the environment's inherent uncertainty about the precise interleave and synchronization between control (Max) and signal (MSP) events.

My increasing involvement with formalism and purely symbolic algorithms naturally led me to concrete applications resorting to "notes". The

[7] Some of them, incidentally, were former students of mine in the Lyons Conservatory.

most obvious consequence of this shift in my personal interests was thus to bring me back to a *music of notes*, away from the *music of sounds* which had been the natural emanation and consequence of my involvement with computer sound synthesis and processing — as well as with analog electronic music. In the first case, the so called *note* entity constitutes the musical *atom*. Music is performed out of collections of such atoms, and finds its very motivation and existence in the established, and hopefully perceived, interrelations between these atomic entities. In the second type of music, schematically, the atomic entity is infinitely reduced to the mere sound sample — e.g. 1/44100th of a second. Incidentally, one can deplore the fact that usage has not distinguished between the two acceptations of the word *sample* in computer music jargon, which can designate a numeric — thus also highly symbolic — representation of the instantaneous amplitude of a sound wave during 0,000022675736961 sec., as well as a recorded sound fragment lasting in the order of a few tenths of seconds or more — and typically used by *samplers* — which is truly much nearer to the concept of note.

To a certain extent, one could say that, in a music of sounds, the music resides in the sounds or sound textures themselves, in their very existence throughout the elapsing time, before inter-relations between sounds or sound textures can be established. In such music, the time scale at which the musicality will reveal itself is not predictable — or often a surprise in itself. Such music thus needs a relatively long time for the listener to understand and construct its musicality.

On the other hand, in a music of notes, the music happens immediately *between* the notes, in their dynamic interrelations evanescently established over the elapsing time. With very rare exceptions, notes bear absolutely no musical content in themselves — not more than single isolated digital audio signal samples. They are taken as abstract entities, and consequently have the advantage of allowing an optimal display and rendered perceptibility of their interrelations, on a very fine time scale. Helped by cultural references, the listener is thus nearly instantly able to construct some musicality.

As a consequence of this difference, for instance, Pierre Boulez points at the obviously problematic existence of *polyphony* in music of sounds, compared to our long and sophisticated western polyphonic traditions in music of notes. In music of sounds, polyphony is often reduced to the concept of *layers*.

At first sight, formalism and symbolic computing do not seem to be applicable to create or control digital audio signal — i.e. to the sample atom mentioned above. But there have been tentatives, by Herbert Brün, Iannis Xenakis, Paul Berg, among others. I myself, in the 1970's and 80's have dreamt of universal musical processes, creating musics of sounds out of unique principles fractally applied at all scales, from the macro-form down to the atomic samples. My piece *Les portes du sombre Dis* partially attempts this, only stoping at the level of a synthesized atomic *grain* — itself implemented over an extended scale of durations, ranging from a few milliseconds to a few minutes. Unfortunately, I have never neared an integral realization of the totalitarian approach I was thinking of. But... is it really unfortunate? The only certitude is that the results would have been typical experimental musics of sounds.

Although nothing guarantees the musical validity or interest of such experiments, one thing is absolutely certain: that a computer could modelize them.

The computer is an universal, or generic, automaton. The automaton of automata. Much more than a calculating machine. Some languages have the advantage of using a more appropriate word to name this automaton: for instance the french *ordinateur,* spanish *ordenador,* etc., which was first introduced by IBM France in 1955, following a suggestion of philologist Jacques Perret. This term originates in the abstract concept of *order* — typically: taking decisions about situations and things, arranging them in relation to each other, etc. — and is potentially open to any sort of symbolic manipulations or operations. The english *computer*, which originates in accounting terminology, and has been transplanted in many other languages, or the german *Rechner* for instance, are etymologically more limited to the concept of arithmetical calculation[8].

The possibility for computers to modelize anything whatsoever, from music to climate changes, even to future generations of computers, constitutes a major scientific breakthrough of the XXth century. These automata open new perspectives and allow to explore the potential of any idea, in all domains. The optical discoveries made during the late 1500's and early 1600's, which led to the telescope and microscope, seem to me comparable. Of course, these new optical tools did not *change* the world, but completely *shifted our vision* of the world, allowing to discover unsuspected aspects and dimensions of the universe, opening new paths for exploration.

---

[8] Literally: counting with *calculi* = latin for *small pebbles*.

As the ultimate consequences of the breakthrough provoked by these dutch optical experimenters covered a considerable time span — for instance from Galileo in the beginning of the XVIIth century to Pasteur during the second half of the XIXth — it is most probable that computers will bring forward unimaginable surprises and discoveries in the future.

The simple two state unidimensional cellular automata popularized by Stephen Wolfram are a striking example of the potentiality of computer modelization. In principle, some ancient egyptian or greek shepherd or philosopher could have imagined such pastime "games" using a simple stick tracing checkered patterns on moist sand. Clearly, this is not impossible. However, he would most likely have been bored out and would have walked away from his drawings before visualizing any interesting patterns and seizing the potential of his invention. Nowadays, computer implementations have made of these automata one of the obvious illustrations enlightening the most important modern concept of complexity.

The fact that computers allow to modelize, and eventually realize any musical idea, completely modifies the ways and means of musical composition. It remains of course possible to do without, but younger generation composers mostly integrate computers in their daily musical tasks. At most as a generator of the final musical result, or at least as a modelizing and testing benchmark. As computers have completely invested our professional and personal environments, there is no reason for an exception in musical creation. Throughout nearly sixty years of evolution of computer music, computers have become, and are now, clearly, versatile tools prolonging our musical thoughts and imaginations. In the end, only music remains. Music itself.

Any segregation between art, science and technology is nothing but sterile blindness. As shown by the progress of the increasingly virtual and abstract fundamental sciences, human destiny is always ultimately confronted to the same profound interrogations. Are music and mathematics not the products of the same imaginations? Of the same brains? For unknown reasons, they both merge with essential structures of the universe. Indeed they both stem out from this essence. Why should contemporary technologies belong to a different nature than contemporary arts, since they are the product of the same history within the same bodies and minds? Humans secrete technology, and science, and art. We can not even avoid it. That is our very definition. That is our nature.