# Towards the roots of computer-assisted composition: Exploring one thread.

*An Interview with Paul Berg*

**Ángel Arranz**
Institute of Sonology
Juliana van Stolberglaan 1
2595 CA The Hague – The Netherlands
angel [at] angelarranz.com

The current article is an interview made to Paul Berg the 18th of May 2009 in the Computer Studio at the Institute of Sonology of The Hague – The Netherlands. This text scrutinizes some Paul Berg's personal investigations as a composer and programmer, revolving around the origins and the emerging of some of the first hardware applications in terms of computer-assisted composition programs. Subsequent developments on programming, improvements and technical approaches are also outlined as a sort of historical statement of the evolution of such a compositionally-oriented technology within the Institute of Sonology from the seventies until now. This vast formalization concludes in the AC Toolbox, an algorithmic computer-assisted program created by him along the last decade. Some items related to this application, as Clarence Barlow's procedures, G.M. Koenig's and Iannis Xenakis's were discussed too in detriment of a more detailed description, picking up some of the possibilities of the program.

Keywords: *Paul Berg, computer-assisted composition, DA converter, Yuji Takahashi, statistics, Illiac Suite, Barry Truax, FM real-time implementation, Automatic Sound Programs, Herbert Brün, DMX-1000 processor, AC Toolbox, organicism, beta distribution, Clarence Barlow, rhythmic indispensability, Koenig, Projekt 2, selection principles, Xenakis, probabilistic functions, data sections, French musical tradition, Poisson function.*

## Beginnings and first discoveries on incipient technology

> *AA: Before your arrival to The Netherlands, what was your background -as a composer, technician, engineer-, what was your previous formation?*

PB: [*Long and deep pause*] Hmmm... I never talk about my previous stuff. But originally I was an organist. When I went to conservatory to study organ, in the very beginning, like within the first few weeks, Xenakis came to visit to my conservatory, and that changed my life.

Secondly, at about the same time my conservatory got a grant from one of these foundations with a lot of money, I cannot remember which one -Rockefeller Foundation or something like that- to take a DA converter to the computer that was at the basement of the physics building, and musicians could use the computer at night for midnight to three in the morning. And that was very exciting, because it was like the first music school in America that could do that. University could not do that, because it was not a musical school, and we could do that at the middle of the night, and now that was very exciting as well. And that is how this started. And I would say maybe those two things were really important; one, Xenakis came to visit, talked about being and becoming and harder other stuff -you know, the school orchestra tried to play some of his orchestral pieces, which of course they could not, but that was exciting-, and Yuji Takahashi came with him, and that was quite amazing too, because he is very gentle person; at least he appears on stage he is a very gentle person. He banged the piano a lot, then they brought out another new piano, and he banged on that new piano a lot, and so on and some forth. So it was a very exciting event. So, on one that hand it was a sort of musical aesthetic inspiration all context, and the other thing, that it became possible as well.

But my past it was not as a technician. I am for the generation where composers and musicians, whatever it was what we called ourselves, programmed just because we had to. Because no one else was doing it for us. We had to do ourselves. I mean, you could not go to the store and buy a program, and you could not even... there were not that many programs anyway. This is the end of the sixties, beginning of the seventies, people tried to decide what music should be -when one would have this possibilities-. So, everyone had an answer. That answer you had to do yourself, basically. Or if you were famous, you could have slaves that did it for you. I was young, so that was the way we had to do it ourselves.

The place of my conservatory was part of a small college, which had two parts: they had a conservatory about five hundred people, and they had a college of liberal arts of two thousand people. And that means why was a physics department, which had a computer in that conservatory, could put a DA converter on. So that was very exciting, and well, also that means that you could take classes like statistics, and in statistics class just you said: 'Ok, at tonight we are going to program the computer', not just like explaining what we are going to do. That was all this idea, and that was also very informative. So, those kinds of things played a role.

"I am for the generation where composers and musicians, whatever it was what we called ourselves, programmed just because we had to. Because no one else was doing it for us."

They were other kind of events start circling around. My study was very near Cleveland, and that time Boulez was the principal guest conductor of the Cleveland Orchestra. They were coming to play around school, he becoming director school orchestra. That was announced in spite of it: 'Xenakis is coming in'… and Xenakis did not conduct! [*Laughs*]. Those kinds of things played a role in my background. I never was a technician, and I never studied computer science or anything like that.

In general, for a long time -and maybe is not longer too- that was a time actually that for certain kind of things musicians were much more further along than computer scientists, because musicians work with time, and for computer scientists then conception of time was actually quite limited. I mean, people who was making medical software in real-time, but that was like measuring heartbeats, which is not that fast. You have not the 44 thousand things or 20 thousand things within a second. It does not go that fast, so musicians all we were developing techniques that others just not yet do. And that also why the musicians have to do it themselves. Another famous example of this is that Hiller made this program to plot… I think it was piano piece, maybe is about another kind of thing I do not remember...

*AA: Before the Illiac Suite?*

PB: No, no it was after that, I think when I was in Buffalo. That plotter had been bought for scientific purposes. This is the first time that anybody plotted so much, and something so long, and they found out that the software did not work at all! So this is again, musicians have sort of demands: it is going to be more, it is going to be faster, and it has to be more this… So we had to do ourselves, because not even the scientists were doing it for us. That was a certain time.

Now there is all this software, concepts and things that you can complicate maybe for musicians understand, I do not know. Then it was not choice. That was the reason about why Xenakis programmed, Koenig programmed, every buddy programmed. At the surroundings of the institute also, most composers I think programmed. At least it is the one that they did really interesting things. That was just a normal part of the activity, because otherwise, what does it damn more you could do if you did not program? It is like making your own brushes, mixing your own paint and writing your own programs. Computers were so complicated then... so, it was less of the problem. In the colloquium of this past week, Casper [*Referring to the Dutch composer and a current sonology student Casper Schipper*] studied this piece about Cort Lippe, 'Vosive', which is from the early eighties, and that was programmed. He programmed it, I mean [*Knocking the table*]: it had to be done as most things were in the seventies.

"At the surroundings of the Institute also, most composers I think programmed. At least it is the one that they did really interesting things."

**Automatic Sound Programs:**
**Development and computation of musical automata at different compositional levels**

*AA: What was your creative activity here and how did you start to work at the Institute? How did it change your viewpoint about music?*

PB: Ok, that seems to me that maybe will be a couple of questions [*Laughs*]. One of this is how my viewpoints about music changed, and the other one was... how they were doing it?

*AA: Yes. Summarizing both, how did you synthesize your background with your new prospective here?*

"There was a computer and people was centered around a computer, but also around ... ideas."

PB: I do not know if I ever did it. I am from a generation where people were curious. And people were interesting in a large number of different topics. Maybe then there were people like that too, but now you see people hardly interesting in making sound using SuperCollider, and that interest is not further than that. There were more people interested in different things, because maybe it was not quite this clear about what you could do with this kind of technology, and what you should do with this technology, whatever is going to turn into and so on and so forth. When I first arrived at Sonology there was a computer. That is a big difference: look at the number of computers in this room. [*Pointing to the old computer pieces within the shelves alongside the back wall of the room*] ... in that thing! [*Laughs*]. That was a really different thing. I mean, you had to go to someplace, where there was a computer and people was centered around a computer, but also around ... ideas. You could not spend all week on the computer, because there was only one computer. I did not know, when I first arrived at Sonology, that I was going to end up doing what I ended up doing. Nor did I maybe even know that I wanted to do that.

There were a number of different kinds of things. For instance, when I arrived at Sonology, Otto Laske was still there. There was all theory, and for a while I found it very fascinating. Of course, Koenig is very fascinating in terms of a certain aesthetic context. He talked about a lot about technical issues in the days when I was there, because he was involved in figuring out how to solve some technical issues; really technical issues, not compositionally technical issues, but really technical issues.

And then, at a certain point -and that was not like a decision that I sort of woke up one day and knew, because one seldom does-, I think that I maybe started to regard it as technical issues, I do not know. But then it turned into something fundamental to look at. Among other things, I was into the city and I was indeed trying to see how you get sound out of this computer. There were actually those many programs at that time in the Institute that worked, there were three programs by Barry Truax [(1)] and one of them would use frequency modulation. I used at that time that old one [*Pointing again with his finger*], which really worked. I used it for a while learning things.

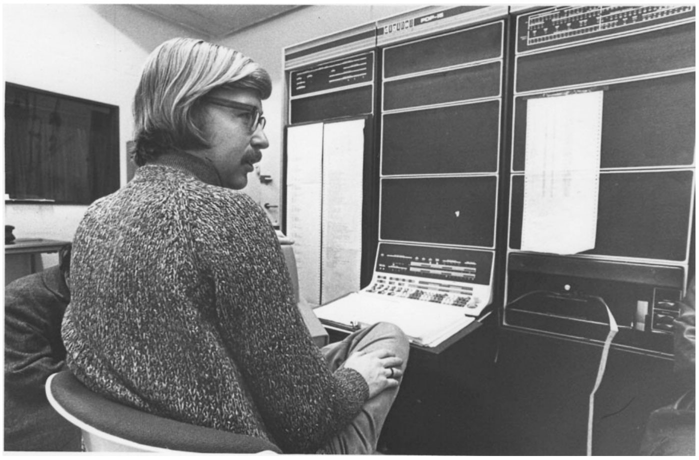


*Fig. 1: Barry Truax at the Institute of Sonology in 1973, sat in front of the PDP-15 computer, nowadays exhibited at the Varesezaal, Koninklik Conservatorium Den Haag (source: http://www.sfu.ca/~truax/).*

“What do you have to do to make sound? What you have to do is to make numbers really fast.”

Then I decided to know more about programming at a nitty-gritty level, so I could do something on my own. To that extend, at a certain point it became obvious to me what it is what I wanted to do. And that was ignoring science. And pay no attention to science, and pay no attention to other scientific insights, and instead say: What is it that a computer can actually do, if you want to make sound, or what do you have to do to make sound? What you have to do is to make numbers really fast. You do not have to solve equations that produce filters, you do not have to read waveforms, and you just have to produce numbers. If, for instance, you were making a digital recording at Beethoven -at that time we could not do that at the Institute, because we have not any place to store that data at Beethoven-, once you read it into the computer it was just numbers. My idea was: ‘Well, ok, we forget reading Beethoven into the computer and just make the numbers’. And once you said 'making the numbers’, you can come up with all kinds of compositional strategies for doing that. And you can say: 'Because those numbers are parts of waveforms, and are parts of sound pressure functions, which continuously change over time, then you compose on that very micro-scale'. That was very exciting. It was exciting because something happened and sound came out. It is exciting because working in real-time within a very small computer you could be doing these things. That was exciting because you had to develop techniques to control and something like this. So you have the micro-level, you have to make something with the middle level -the meso-level, as people now like to talk about that. So, there is lot of, let us say, compositional motivation to the way that sound was produced. And then, the music was produced like that. Ok, that was not something that I thought of ahead of time. At the same point, the decision by which I would forget science and I just was going to make numbers, that was conscious in the fact that in retrospect you could say that maybe the reason why this worked, to a certain extent, is because it was compositionally motivated and it was about change. Because if it does not have change, then it is just white noise -which is fascinating, but there is a limit how long you can listen to that, probably-. You have this kind of thing and there was something noisy about it, which is ... [*small pause*] one the basic ingredients of electronic music, in any case.

So, that led to a number of programs, which I did put together into a library and gave them a name, tied a ribbon around it and called it ASP (Automatic Sound Programs). All you can do is turning on the button and then they will run. You have not other control, so they just did it themselves. Even if you do not like it, you can turn it off! [*Laughs*] I have seen a precious scene in those computers in the Varesezaal [2]. They have that switches, so you could put a number on those switches [*Exaggerating and gesticulating with his fingers on the table*], and that is where we started. You have not idea what that meant, but typed in a number and turned in on and it will go, and ... forever!! I mean, literally, forever! If the computer was still working, one of those programs could be still running today after thirty years, they would just go on forever. And you can see that output as being a composition. You just turn around to another, tape recorder and turn the button off: that was it. You can say that it is something that generates moments -if you take these moments and you can add it to them to a piece, or something like that. Or you can say it is just a generator, like a sine generator or a saw tooth generator, only that it is a more complex kind of generator. But you know, it is one of those things that you turn on and it just goes on forever.

Those programs were really short, because they had not to do science, or worried about frequencies. Because frequency was not a parameter, duration was not a parameter: none of those things were parameters. So, that means that the programs could be rather small and efficient. What you heard -as well you could try to describe it in terms of frequencies or durations- it was basically noise, some shape or form.

> "Actually, everybody in Sonology worked in real-time, just because there was not place to store things."

That led to another series of programs, which I learned a lot by doing that, and I actually saw a certain relationships among those programs. I could not really predict, when I was writing one of those programs, what it was going to sound like. I always have been influenced by Herbert Brün. He basically made that distinction between producing a system and listening to what it sounds like, or trying to imagine the sound and then program that sound. And he was for the first method. Describing this process in this system with this compositional idea and ... 'Oh, that's what it sounds like!' As opposed to: 'I want to have a canary singing in there', whatever it is. So, that was not such a problem for me since... But it also makes it very hard to explain these things to other people, because this is doing all this weird stuff, changing the sample wave for every sample... who knows.

And since I saw certain similarities in the things, then I liked it to turn into a language. There were some processes that came better and language gives you more possibilities, and finding those more possibilities you can easily make other things, but you can also explain it to people and talk to people. And that was the generation where people still liked to talk to each other and share, and so on and so forth. So, that was an important thing. But the idea was still basically the same: producing numbers, and producing them fast enough, so that some kind of sound came out. And that, again, worked in real-time: you turn it on, and then it produces it.

Actually, everybody in Sonology worked in real-time, just because there was not place to store things, so that work that people were doing in MIT and places like that was very different from the most happening in Sonology, because Sonology was doing real-time sound production. This is the place where the first real-time FM implementation happened - Barry Truax did that- and everything else was working in real-time. It was not about performance, because that computer, even though it was small in terms of processing power, it was physically big. If you move it some place, it probably takes you three weeks to get it running again or something, so it was in the studio and there is where it was.

In terms of the methodology, it was happening in real-time. So, that meant that people made pieces, either they collected material and later mixed it some place else, or they produced the pieces bit-by-bit into a computer program. And then, once you have that whole thing, then it is turned on and it comes out. Each section of each part has been very carefully constructed. But since random numbers played a role in this -and if you give it the same random start numbers, you get the same results-, you can always reproduce what you are doing: you set up the name, you give it some initial conditions and then you listen. That also means that listening was a very important part of this. And that was Sonology in the seventies: listening. That was very important. As opposed to say people that were interested in building robots. You have these machines, do something and ... 'I can write music too!' [*Imitating a monotonous robot voice with a little bit sarcasm*] It was not about that; it was about this process of people listening.

That turned into at a certain point doing things at a slightly different level. There was a project started within this institution, a co-operate project between Sonology and the Conservatory, and one of the things was to set up a computer studio, and what we did for that is we bought a DMX-1000 single processor [(3)].

*AA: Is it in this rear..?*

PB: That thing like there! [*Pointing again to the shelves of the Computer Studio*]. It was a programmable device, so you can program it to be a bunch of oscillators, program it to be a bunch of filters, program it to do FM or program it to do all kind of things ... in real-time! So, one of the things that I did is: I took something similar to the software that I had done, in terms of the language to produce numbers. And instead of it, I said: 'Ok, now the output is not about producing sample values, but it is produced to control information for those things'. So, I could program now to be a bunch of oscillators in the new use that approaches to generate frequencies or sine waves, or filter parameters, or whatever! There was a certain kind of similar interest in process, only the level of the application was not the micro-level; it is more, let us say, this middle level. And that again was the 'language-kind-of-thing'. It was a language: you could write a program in, and then compile it and run it. And again, you can describe an entire piece based on that program, but now they content all the patches needed to do in it, now it happens in real-time, or you can gather the material in those some places and edit it.

And going from there, the next step turned into, well, going from this middle level, controlling programs that in enough themselves could produce sound. A code, for a sine generator, does not produce sound by itself. You have to feed it with the information. But we had slightly higher level. Then, the next step that was made from that was to forget the sound generation and make code that produces data that could be used for, I call them, 'traditional mechanical instruments', such as piano, violins, things like that. And that ultimately resulted in AC Toolbox.

*AA: That is the point.*

PB: But also the AC Toolbox now is becoming very interesting in controlling sound processes again... But this is the sort of trajectory of the course of a couple decades, which was not planned out ahead of time, but started very much in this micro-level. And then it made this step up to controlling these other processes. And then, for various reasons, became these interesting other things. In all these things -on one hand you can give a justification in terms of aesthetics, and now it will be partially true, and on the other hand you can give a justification in terms all kinds of accidental occurrences. For instance, that I took this step to the micro-level to the middle level had everything to do with the fact that I was involved in this project here to set up the computer studio and bought the DMX-1000, with which I could do that. It was not bought so I could that: it was bought to do various things. If the fact had not happened, I might not have made that particular step. And that step more interesting in instrumental possibilities -in terms of, again, traditional mechanical instruments-, not only had to do with this continuous level of abstraction, which you are seeing in this path -these tiny little programs, to a language making silly old sounds, to controlling simple little sound processes, to having oboes playing these things.

At a certain point in the eighties, I bought a computer myself at home. And then computers… you know, in those days you could not really make sound with them. You could hook up midi devices, but you could not really do sound as a... they just were not that powerful, but you could do notes, that worked really well! On one hand there is this logical, philosophical, musical aesthetical line, and on the other hand there are some of these practical things that play their role as well. And the logical line is something you can maybe see it looking backwards, like as I just can look it backwards now and say: 'Well, Xenakis came to my conservatory, and the conservatory bought a DA converter'. Everything flows from that. But that was not obvious at that time. That path and that development could have ended up differently at any of these moments, so who knows? On one hand is causality, and on the other hand it is random. If I had not ended up in Sonology, I do not think I would had been doing those kind of things someplace else, because partially, it was caught up with the fact that Sonology had this computer, had a hardware random number generator that I could use it and I was interested, and there was the stimulating environment ... all those kind of things! So, on one hand there is a line and in the other hand that line is sort of random, even though it looks to be straight now!

## AC Toolbox and some of its attributes: Barlow, Koenig, Xenakis

> *AA: Ok, those were very long answers, so that is fantastic because it invites me to let the circle close itself. Now a very simple question: what is AC Toolbox?*

PB: Well, I do not know if that was a simple question. I mean, on one hand there is a simple answer. The simple answer is: it is a software environment where data can be calculated, and using tools that might be ascribed to a certain kind of composer or a certain groups of composers. What that data can be and what it could be used for and how it could be represented is different. You can just calculate numbers, you could calculate MIDI data, you calculate the kind of text-based score, you calculate MIDI files, you calculate data for CSound, you calculate data for SuperCollider, so on and so forth. So, on one hand it is just what it is. It is just programming, you can do those kinds of things. It contains a lot of tools, some of which conflict with each other. It has tools so you can examine ideas on Koenig, but also tools so you could examine ideas from spectral composers. All this stuff is made in AC Toolbox; people could explore things and try things out and make music, or make parts of the music. That was one answer to what it is.

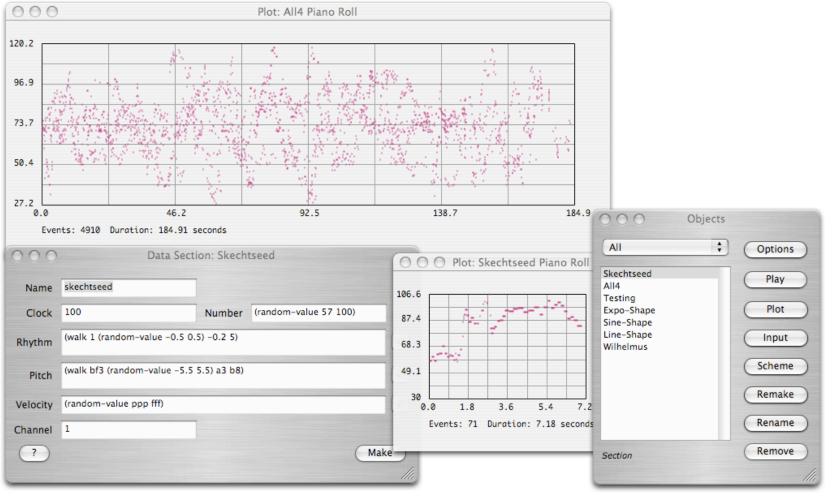
Another answer to it is just another manifestation of my curiosity about a number of things, try things out, try to look at some of these approaches that people had towards composing, using probability, using spectra or using chaos or something like that. It is an environment that I can try these things out and realize them.

*AA: I would like to introduce an insert here. I have the impression when I use AC Toolbox –I confess I have used a very small part of the program, perhaps less than a 1% of the real possibilities-, I have the impression then that this programmed deconstruction of yours is pretty organic. You can select whatever different bank of tools or, let us say, virtual spaces in order to work with, and on the other hand you feel the hierarchy of different settings of the program, and at the same time you feel that it is organic, that it is done by using different layers on time. You feel that the program is still growing up. I am very curious about what is the core of the program. What were the first pillars? In which directions the program began to grow?*

PB: It was very organic, and the intention was not to make a program –and the intention was not really to make a program that other people could use, it is just that I needed some functions to do some things. So, I programmed them for myself to be able to do that, and in the course of doing that I gathered a collection of various kinds of functions that were useful. There were some other things that I programmed for myself at that time and I threw away for various reasons. I was working on a piece for harp, and turned out to my disappointment that the harpist thought that I would indicate the pedaling –I thought this is more something for harpists to do themselves-, and since the pedaling was sort of complicated so I said: 'I could write a program to deal with the pedaling, and if it did not work out, then just change the notes'. That was thrown away. I did not think that it was a sufficient general use to include it. For my own personal use and the things I was using, I had this number of programs. They were written as text programs and they had not a particular interface, except the kind of interface you have with any text-based programming language.

These tools were based on solving issues that I had about things. And then, at a certain point, I tried one day to make dialogue boxes, as a means of putting the data in. I was nothing behind that idea of 'Oh, if I use dialogue boxes everybody could use those. Ok, let us try to organize it in the same dialogues boxes'. Actually it is quite clear. And that clarifies certain things for me. And that started that path of thinking that maybe I could organize this in a way less personal, get these things together, turn it into a program, gather up some of these tools and have a certain method for presenting this information, and so on and so forth. Things just started going from there. In terms of the very beginning thing, you could say the first thing is that I programmed for myself… *[Interjecting]* was a beta distribution! I can remember needing that for some oboe part in a piece, and that was very useful. I do not think that the oboe player liked it particularly [*Joking*], because there were issues, I understand that, so that might have been the very first one of those things. In terms of the AC Toolbox, they maybe were the first kinds of things. It was not even called AC Toolbox in that time.

"It was very organic, and the intention was not to make a program –and the intention was not really to make a program that other people could use, it is just that I needed some functions to do some things."



Events: 4910 Duration: 184.91 seconds

Data Section: Skechtseed

Name: skechtseed

Clock: 100

Number: (random-value 57 100)

Rhythm: (walk 1 (random-value -0.5 0.5) -0.2 5)

Pitch: (walk bf3 (random-value -5.5 5.5) a3 b8)

Velocity: (random-value ppp fff)

Channel: 1

Events: 71 Duration: 7.18 seconds

Skechtseed
All4
Testing
Expo-Shape
Sine-Shape
Line-Shape
Wilhelmus

*Fig. 2: implementation on AC Toolbox of a seed by Ángel Arranz within the Data Section of the program. As the result as a bigger section were plotted on both timeline drawings.*

Then, these issues turn out to be Data Sections. You would take certain rhythmic procedures and pitch procedures that they did not necessarily have to do with each other. They could be joined to be a thing. I think that was maybe the first combination of those things: I did something for pitch, I did something for rhythm, which happened at the same time, and I could deal with that somehow. And dealing with it somehow –including that the oboe players and she could not play it like that *[Simulating to touch buttons on the table]*, we changed it. So, at a certain point I thought that it might be useful for my classes: we would talk about these things, and then it might be useful after some documentation. But actually it turns out convenient for me too, because, I need the documentation: I forget how things work too.

To the extent of the whole thing is organic, it maybe is because is something that basically one person made. You know, as opposed to a committee. And a lot of software is made by committees. Committees never talk to each other and just send e-mails to each other. This is about me and how I see these things at different points. Sometimes I think this is more important, sometimes I think that is more important. But that is a certain development to it.

*AA: There are some special insights about a couple of composers, maybe more. One of them is Xenakis. There are some specific tools about some Clarence Barlow's techniques as well, for instance the rhythmic indispensability, or Gottfried Michael Koenig, who is very present too. Let's focus them a little bit. Could you describe briefly how these implementations were done, in case of Xenakis? Concretely, what Xenakis's techniques did you implement?*

PB: [*Deep sigh*]: Ok, let's start with the easy things first. You mentioned Clarence Barlow's idea about rhythmic indispensability [4]. That is relatively straightforward, because he has talked about that a lot, published about that, even though his formula was not always right in some of his publications about that, but it is an idea and you can look at it, understand it and maybe find it interesting. So, that was merely: 'Ok, this is what I think it should be, so this way I am going to do'. But then there were some things, which I did not understand –or Clarence maybe did not express it clearly, or both- and then I just did my own interpretation of that.

The basic idea of the indispensability is to assign weights to different parts of measures in order to give the impression that you have different subdivisions of a space. So you think that is 12/8, whatever it is. I took that basic concept and the only way you can tell that, and in Clarence's implementation too, is by throwing away things. By having holes you could realize what is important. I mean, if you fold the beats around, you will have no idea if it is 12, 8 or something else. So that straightforward could be realized in terms of generating the weights in the information, partially because I did not quite understand his formula and partially the reason was -he later told me- that the old version of formula was not right, and he had a better one. But by then I had already come out with my own way of doing that, which was more random. So you can have different possible ways of doing 12/8. It will still be recognizable as 12/8, but it will not necessary be only one thing. And I guest that was maybe an interest of mine. So, that is the reason why it was straightforward.

"You can say a tendency mask could be a generalization of the field concept of Stockhausen in the way to allow transitions. And I generalized that even more by *drawing* a mask."

In terms of Koenig, in Projekt 2 and also in some of his synthesis programs, he had selection principles. You have such whole ideal players; if you specify certain stockpile of things, you make a choice from this stockpile. And from this stockpile you reorganize it in tables and then you make choices from those tables. And those choices are made by selection principles, such as tendency masks coming to play, random-choices and things like that. Those were generalizations: you can say on a tendency mask that is a generalization of field concept of Stockhausen, in such a way that you can produce transitions. On one hand it is a very fundamental useful thing, and on the other hand it is a very simple change. You could say that the field concept of Stockhausen is something like this, which is a rectangle, and a tendency mask is just one of the sides of that rectangle that is smaller than the other, which means that it is going to change over time. That means you could do transitions in this way by having a kind of change, so that was obviously a useful technique. In his group mechanism of choosing values and repeating them, which suggest certain kind of musical activities, it was a very obvious thing to choose. I did not choose list selection tables or selection principles, but the idea that he had on the selection principles. Only in the AC Toolbox I basically divorced them from the context of Koenig, and I said: 'Here we have these selection principles, and you can use them how you want to… it doesn't have to be picking things from this table or that table'.

The second thing is that there is a certain historical musical tradition to what Koenig was talking about. I would refer to that thing you can say a tendency mask could be a generalization of the field concept of Stockhausen in the way to allow transitions. And I generalized that even more by drawing a mask. You do not have these very linear kinds of things, as Koenig specified them: you just draw them. So that is sort of easing up the restrains a bit.

In other one of Koenig's selection principles that was grouped, you can pick up values, either at random or random with repetition check, determine how often it is going to be repeated by random over the repetition check. And I just generalized that: you can pick up a value of any function you have in the AC Toolbox and take the repetition –any value you might have in the AC Toolbox. Koenig made his choice for a certain specific reasons that had to do with the relationship of regularity and irregularity in a certain serial tradition, but I just took a step away from that; but anyhow, it is based on that kind of idea.

Other things are merely random: there is not really much you can change about that. In terms of making random choices without repetition, this is something that you find in a lot of computer programs these days, meaning Koenig's idea was a form of expressing irregularity. Because if nothing is repeated, that is very irregular. In Max/MSP you have an object that does it as well, 'urn'. I don't think they were thinking in terms of musical irregularity. They were thinking more in terms of statistics. But functionally, it is the same.

And also, the irregularity of series, at a certain extent, is a generalization of the whole concept of working with series, because you are making choices from things that are different. It is a generalization. Obviously it is not paying attention to things like tetra-chords, or any of those kinds of issues. People did series with notes; people did series with rhythm… so on and so forth. Koenig said: 'You have some series, I will make random choices all about and not repeated it' –it is an extreme generalization of that kind of thing-. The only difference I did that is putting this context of the AC Toolbox where that could be applied to anything, basically.

In terms of Xenakis, it is more loose, in the sense that people could read 'Formalized Music' all day, everyday, and discover new things. And I suppose one could spend a lot of time trying to take the mathematical models that he describes and implement them. But I chose not to do that. Two reasons: one reason is I do not know how accurate those mathematical models were in the first place. You often heard from people that had too much time on their hands and they studied the computer programs that Xenakis actually wrote, and I have seen a few of these articles from people claiming that is not possible that that program would produce the results that Xenakis did. Nouritza Matossian, in her biography, she mentions things like in 'Atrées' he threw away 25% of the output from the computer anyway, and just made it up. And that is fine: composers do that. But then there is really no point trying to model exactly what is in the book, because Xenakis did not do it like that anyway.

"Xenakis is a French composer, in the tradition of Debussy. And that for me means that it is not about the mathematics."

So, then I was more interested, let us say, in the general idea. And the general idea is that you can specify behavior by using probability functions. You can specify developments in music with these probability functions by having changes at the parameters. As Xenakis describes you could make very drastic changes, just by changing one parameter. The idea that these functions could be applied in different ways was not important. For me it was an inspiration of including a number of these probability functions. Xenakis was very fan of using exponential distribution -and that is in there-, he was also very fan of Poisson -and that is not in there, just because I did not put it in there-. I wanted to program the Poisson function. I could have put it in there, but I just did not. So, it was not about: 'Ok, that is what Xenakis did, and now you can use this and do that'. I was generalizing.

There is something that is important about these statistical functions, and that aspect could play a role. And you can make this kind of transitions and differentiations, such as Xenakis were talking about. And another thing -and again, this is a generalization- is density. In the AC Toolbox there is a dialogue box, Density Section, in which you can define a number of events in a specific amount of time, and you can do that by using statistical functions to fill that up. That is related to Xenakis and this is not. That basic idea comes from Xenakis, but he implemented it in a different way: he had this kind of durations, so I decided to sprinkle these values with these things in there. So, it is related, but it is a very free interpretation. Again, that is an editorial decision on my part to do that kind of thing.

Third thing about Xenakis is that –and that maybe had an influence in terms of where inspiration came for something that I found a way into the AC Toolbox- is that he is a French composer, in the tradition of Debussy. And that for me means that it is not about the mathematics. And that means that it is not really necessary that I try to understand all the mathematics in 'Formalized Music'. Gestural aspect was very important, a certain amount of complexity was important, being able to control densities, being able to control behaviors, and being able to do this in a way in which you could explore things and make different constellations and how you want to do that. And some of those things one could do with the AC Toolbox. It was not an attempt to say: 'This is how Xenakis made 'Atrées', so I am going to do this in the AC Toolbox', that was not the case. Though, maybe one could do things that would result in something similar to 'Atrées', maybe not, I do not know. On one hand that is not even so important to me; on the other hand, I have done things in the AC Toolbox that were sort of similar to what Koenig did in Projekt 2. Just to see if that was possible, but that is more a parallel than an aesthetic necessity.

[1] The programs were POD-4, POD-5 and POD-6. Programs POD-5 and POD-6 were used by people these days, and POD-6 worked with FM. For more information to this respect, visit the Truax's website of the Simon Fraser University in Burnaby – Canada: http://www.sfu.ca/~truax/.

[2] Paul Berg refers to the PDP-15 computer permanently exhibited at the Varesezaal of the Royal Conservatory of The Hague, firstly allocated at the University of Utrecht and finally set within the actual installations of the Institute of Sonology.

[3] More references on the DMX-1000 device can be found them at the article published in the Computer Music Journal, Vol. 3 number 4 (Dec. 1979), pp. 44-49 by Dean Wallraff.

[4] About this Barlow's theory, an implementation exists in Max/MSP as an external object called 'dispenser', created by the composer Georg Hadju. More information about it can be found published by Clarence Barlow at the Computer Music Journal, 11, pp. 44-60, entitled "Two essays on Theory".