WORKSHOP ON CURRENT RESEARCH DIRECTIONS IN COMPUTER MUSIC

Barcelona, Nov 15-16-17, 2001
Audiovisual Institute, Pompeu Fabra University
Revised: August 2003

MUSIC GENERATION PANEL
(A critical review)

Rubén Hinojosa Chapel
Music Technology Group
Pompeu Fabra University
Ocata 1, 08003 Barcelona, Spain
+34 93 542 2104
ruben.hinojosa@iua.upf.es

MOSART
Music Generation ?
Barry Eaglestone
MISTRES
Department of Information Studies
University of Sheffield, UK

Eaglestone's presentation

MUSIC GENERATION PANEL (A critical review)

Abstract

In the frame of the Workshop on Current Research Directions in Computer Music, a **Music Generation Panel** took place. The chair was **Henkjan de Honing** (NICI-University of Nijmegen, The Netherlands), who introduced and conducted the panel, whose members were (in order of appearance): **Barry Eaglestone** (University of Sheffield, United Kingdom), **Roger B. Dannenberg** (Carnegie Mellon University, Pittsburgh, USA), **Eduard Resina** (IUA-Pompeu Fabra University in Barcelona, Spain) and **Jens Arnspang** (DIKU-University of Copenhagen, Denmark). Each of the panel members made a short intervention and, after Arnsprang's words, some questions came from the audience. The panellist's answers were focused on what they have talked about.

To make available the main topics discussed in the Music Generation Panel, to the community of computer science researchers, composers, musicians, students and everybody who is interested in Computer Aided Composition, we have made a summary and have added a critical review at the end. As an introduction to the purposes of this panel, here is the

1. Call to the Music Generation Panel

The abundance of music generation tools and systems is well documented. These range from AI-based systems for autonomous generation of musical ideas to conventional design tools, for example, for designing and rendering of sounds. However, emerging de facto standards have been short lived, generating frustration rather than satisfaction. This panel will focus on why this is so, i.e., the extent to which accumulated results of this effort fail to satisfy the aspirations of composers. Three specific aspects of music generation will be considered. These are:

1 - Representation and contents of the product, i.e., the composition;

2 - The nature of and support for the process, i.e., creativity and composition; roles of artificial intelligence;

3 - Representation and application of individual and community know-how, including the use of repositories and archives to accumulate a history of compositional techniques used, and the use of the Web to provide open access to community knowledge.

Each aspect will be considered from philosophical, conceptual and technological perspectives. The aim is to identify open questions and unsatisfied requirements that technology has the potential to address. Many of these are partly evident in ongoing research in this area. The outcome will form the basis of a proposed scientific agenda for future composition systems research.

2. The Panel

A brief overview of the words by each of the panel members follows:

-Dr. Barry Eaglestone (University of Sheffield, United Kingdom)

(Note: Dr. Eaglestone made his speech based on his paper "Composition Systems Requirements for Creativity: What Research methodology?", so we have kept some paragraphs from that paper, and some words from his speech.)

Electroacoustic music composition tools and systems selectively attempt to provide composers with services they require for music generation, e.g., for accessing, generating, organising and manipulating audio (and other) objects, which constitute the composition. However, a primary aim of composition software also is to create conditions in which composers can be creative in the use of these services. We believe there to be a need to establish a research base for enhancement of support within composition software for creativity.

Research into digital signal processing and the artist's use of sounds is on-going, and consequently, services relating to musical artifacts are volatile and evolving as new techniques and paradigms are integrated into composition software.

The software environment within which those services are used creatively has largely been under researched. Instead, developments have followed those of software technology. Consequently, there has been a move from asynchronous to synchronous systems, and from text to graphical user interfaces.

We believe there to be an inherent tension between principles of conventional software engineering and the requirements of creative composers. This tension can be explained in terms of models of creativity, which is often characterized by the notion of "divergent" as opposed to "convergent" thinking; the later being associated with relatively predictable logical activity and outcomes, the former with less logical and predictable activity and outcomes.

One of the most talked about and most researched area is the one which involves services for creating and manipulating musical artifacts, and there is a lot to say about that. However, we will be looking at the largely neglected area, which is the environment within which those services can be used creatively?

In our research towards this end, we are analysing data collected by observing composers at work in naturalistic settings, using methodologies ranging from software engineering through to the social sciences. What comes out is the tension between the composer's requirements and the conventional wisdom of software engineering.

Specifically, there appears to be a need for an un-typed workspace within which composition artifacts can be freely associated; support which enables composers to control the whole process, employing programming skills at the lowest DSP levels; support for interfaces which challenge the composers' conceptions, rather than reflect them; facilities within which randomness and accidental encounters may occur; and the facility to accumulate both a personal and community repository of know-how.

The future: seeking a definitive composition system is a waste of time. It only generates insatisfaction. Software developers need to understand composers better.

I suggest two worldwide projects to the community:

1- To develop a research base for better environments that support creativity.
2- To work with the community to establish some musical grid, network and know-how base.

Software for Electroacoustic Music Composition

COMPOSITION TOOL   COMPOSITION TOOL   COMPOSITION TOOL

- Physical (Computational / Data / DSP)
- Logical (Tools / Materials / Composition objects)
- Perceptual (Freely associated untyped objects)
- Community know-how (A Grid)
- Personal Know-how (Archives / Knowledge base)
- Composition Support (What? / How? / Where? / Again? / Suggest? / Random?)

(Diagram by Eaglestone)

Jens Arnsprang: I think this is related with education. Composers become craftsmen, following the transition into the other side of the user environments that better support creativity. The long answer when I have the chance will be: new education.

-Dr. Roger B. Dannenberg (Carnegie Mellon University, Pittsburgh, USA)

I would like to propose that the composition has two components:

1- Methodological applications of standard practice.
2- Creative practice, which is anything but methodological.

So, what I mean by methodological application of standard practice are techniques, maybe know-how of good works, including using digital audio, synthesis hardware and software, music notation, publishing software, also organizational materials. We record and use performances gestures and knowledge, and know about simple manipulations, such as: stretching, transposing and copying.

On the other hand, creative practice is where the music really comes from. The first rule is: break all the rules. The second is: whatever you start with, you want to think about going outside the boundaries. So, almost by definition, if there is a standard practice, you have to go outside to do something creative.

Maybe one of the most common things when people talk about creativity, is to combine things in new ways, often in unanticipated ways. So, all of these things work against any kind of methodology.

In the work that we do, one thing is the very strong tendency, especially in computer science, to try to clarify standard practice and implement it, and by doing that, we ignore the creative practice side because that is in the future work. I think approaches in that way are almost useless because it just ignores the most important part. I think computer scientists have made the same mistake over and over again.

I think there are a lot of things we can do technologically to aid the creative practice. One is building interfaces at the right level. We need to build more open-ended systems; systems that can be invoked by other programs, have options such as text input and output, so when we get a creative idea to combine systems in unusual ways, we need some way to communicate. We can make systems more scriptable, so they can do things they were not originally designed to do.

There is also a need to provide functions at many levels. What I mean here is that maybe there is an application for doing some interesting kinds of synthesis. Maybe it is great, but it is an application, and maybe what I need is a plug-in, or maybe what I need is a function library, or maybe what I need is the source code.

The final point is cost of accessibility. I think this is very critical for helping people be creative. Composers and artists are people that cannot go out and buy every piece of technology. We have this revolution of personal computers and the Internet, which give people access to so much stuff, but it is limited by expensive software applications and proprietary software, you cannot get into the source code and do creative things. It would be good for this community and the whole computer industry, to think about ways to enable artists to get access to those things.

-Eduard Resina (IUA-Pompeu Fabra University, Barcelona, Spain)

What concerns me is basically what makes sound become music. Sound is a natural phenomenon, music is not. Music is an idea we impose on top of sound. Basically it has to do with the perception or the ability to perceive meaningful relationships between sonic events.

From the point of view of algorithmic composition, there are different trends. For instance, starting from some sort of mathematical logic that is not intrinsically musical, and then we want to make musical, in some way. This could be the case of fractal composition. Another trend would be just expressing some standard musical knowledge, traditional knowledge like traditional counterpoint, and implementing this into some system that makes, more or less, automatic composition.

I think there is some sort of composition where you want to start from musical intuition, but you want to set your own rules, you want to define your own musical context. And then after that, you want to be able to implement this algorithmically. In this case you don't depart neither from existing algorithm nor from existing musical context. You have to define all of them.

It is true that certain software or certain tools allow more creative things than others. When you try to define a new musical context, a new set of rules, you take for granted that this set of rules can be implemented in some way; then you want some software that allows you to define new musical contexts. One of the main problems is that it is very easy to get lost with details when you have to be defining every step, every single thing in a composition. It happens quite often that you lose the whole idea of the musical composition, which is essential for the composer. You have a global idea, and you don't want to get lost.

One of the main problems with existing software is that, sometimes, if you have to be really powerful, you have to get down to the small details, and then you really get lost about the whole thing you are trying to work out. In general, they are not very intuitive at all for traditional musicians, who basically come with knowledge or learning of many years in musical terms, in musical concepts, and software quite often does not reflects that.

It would be essential to develop software where you can really work with musical concepts. Software has to be more intuitive for musicians, and certain solutions have to be found in this direction.

-Dr. Jens Arnsprang (DIKU-University of Copenhagen, Denmark)

(Note: I'm sorry, but I was unable to understand every word by Dr. Arnsprang. The main idea of his speech follows.)

I suggest new education, a new kind of education.

```
        Art – Computer Science – Multimedia
             \           |           /
              What kind of education?
```

3. Our Final Remarks

Finally, we would like to make some critical comments. Firstly we disagree, to some extend, with the ambiguous use of the word "creative" made by Dr. Eaglestone. Implicitly, he divides composers into two groups: "creative composers and not creative ones". We think composers are creators by definition. Every time a composer writes a composition, he makes something new, for him and maybe for the rest of the music history. He **always** creates an "object" that never existed before: his music composition. This music shares common elements with others, but **always** has "something new" that makes it different from the rest of music created before. When this "something new" is so little, is what Dr. Dannenberg calls "methodological applications of standard practice". When this "something new" is not so little, is what Dannenberg calls, in a careful way, "creative practice".

So, we would prefer to think about composers who explore new ways of music composition and lead towards new aesthetic concepts, and composers who keep themselves, more or less, in the tradition of music composition. This is, to our mind, what Eaglestone means when he talks about creative composers and, implicitly, about not creative ones.

On the other hand, and thinking in the same direction, we would like to comment the phrase: "*create conditions in which composers can be creative*", and others very much alike.

We depart from a question such as: is there any environment, composition software, tool, etc., where a composer cannot be creative? Most music composed throughout history has been written with a pen. With this single pen a lot of composers have made contributions to music development. Great music compositions have been written, and new aesthetic concepts have been explored and developed using only a single pen.

So, should we accept that there is an environment, composition software, tool, etc., where a composer cannot be creative? If we should, maybe is time to tell composers: "forget computers, you cannot be creatives with them, go back to pen and paper times".

This happen because creativity does not reside in any tool, creativity is owned only by the musician who uses that tool. We would prefer to talk about "create conditions that **stimulate** composers' creativity" and support different and open ways of handling musical objects. The limits of creativity are in the mind of the person seated in front of the computer; nevertheless, as Resina said: "*it is true that certain software or certain tools allow more creative things than others*".

"*The future: seeking a definitive composition system is a waste of time. It only generates insatisfaction*" (B. Eaglestone). If we think about a marvellous composition system with the amazing ability to create all kind of music composition, from the past, the present, and even from the future, we totally agree: that is impossible.

Throughout the music history, composers have developed techniques, most of them algorithmic procedures, to handle all the elements of music, say: melody, harmony, rhythm, timbre, articulation, form… What a composition system does is to apply these techniques, and even new or personal ones, to the material provided by the user, i.e., the composer. The ways to handle the elements of music are so many, almost infinite so, from our point of view, it is impossible to find a definitive computer composition system.

On the other hand, as we said before, creativity does not reside in the system, but on the composer who uses that system. We could try to model every way of music creation, and implement those models into a computer system, but who assures there will not be a composer who invents a new one?

"*Software developers need to understand composers better*" (B. Eaglestone). This is a plain truth. From our point of view, the unique way of achieving that is to learn the basics of music composition, and to work as close as possible with composers. If we want to develop medical applications, we should learn the basics (and maybe not only the basics) of medicine related to the software, and work with doctors. If we want to develop applications for astronomy research, we should learn the basics (and maybe not only the basics) of astronomy, and work with astronomers. And of course, if we want to develop music composition applications, we should learn the basics (and maybe not only the basics) of music composition, and work with composers. That is all. To understand composers better, we should think the same way they do.

4. Conclusions

The aim of the Music Generation Panel was to identify open questions and unsatisfied requirements that technology has the potential to address. We think this aim was achieved to some extend. Panellists clarified some theoretical aspects and proposed some directions for future research. We would like to extract what, to our mind, were the main requirements that music technology has the potential to address, and what should form the basis of a proposed scientific agenda for future composition systems research:

Barry Eaglestone: *We believe there to be a need to establish a research base for enhancement of support within composition software for creativity.* (As we have remarked before, "to create conditions that **stimulate** composers' creativity and support different and open ways of handling musical objects".)

Roger B. Dannenberg: *I think there are a lot of things we can do technologically to aid the creative practice. One is building interfaces at the right level. We need to build more open-ended systems; systems that can be invoked by other programs, have options such as text input and output, so when we get to creative idea combining systems in unusually ways, we need some way to communicate. We can make systems more scriptables, so they can do things they were not originally designed to do.*

Eduard Resina: *It would be essential to develop software where you can really work with musical concepts. Software has to be more intuitive for musicians, and certain solutions have to be found in this direction.*

We hope our work would be useful. Please, feel free to send any feedback, they are welcome.