“Leçons” : an Approach to a System for Machine Learning, Improvisation and Musical Performance

Lindborg, PerMagnus

2004

Lindborg, P. (2003) “Leçons” : An Approach to a System for Machine Learning, Improvisation and Musical Performance. Lecture Notes in Computer Science, 2771, 210-216.

<https://hdl.handle.net/10356/81863>

https://doi.org/10.1007/978-3-540-39900-1_18

© 2004 Springer Berlin Heidelberg. This is the author created version of a work that has been peer reviewed and accepted for publication by Lecture Notes in Computer Science, Springer Berlin Heidelberg. It incorporates referee’ s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at:
[http://dx.doi.org/10.1007/978-3-540-39900-1_18].

Downloaded on 26 Apr 2023 21:44:42 SGT

“Leçons”: an Approach to a System for Machine Learning, Improvisation and Musical Performance

PERMAGNUS LINDBORG

www.notam02.no/~perli, permagnus@noos.fr

ABSTRACT

This paper aims at describing an approach to the music performance situation as a laboratory for investigating interactivity. I would like to present “*Leçons pour un apprenti sourd-muet*”¹, where the basic idea is that of two improvisers, a saxophonist and a computer, engaged in a series of musical questions and responses. The situation is inspired from the Japanese *shakuhachi* tradition, where imitating the master performer is a prime element in the apprentice’s learning process. Through listening and imitation, the computer’s responses get closer to that of its master for each turn. In this sense, the computer’s playing emanates from the saxophonist’s phrases and the interactivity in “Leçons” happens on the level of the composition.

COMPOSITION PROCESS FORMALISED

The architecture of “Leçons” takes as point of departure the information exchange between two agents in a conversation-like situation as described in linguistic theory.² Communication between a musician and a machine may be considered to have aspects of both natural and formal languages.

PROCESSES OF COMMUNICATION between two agents

(after Chomsky, 1980-1995)

As Noam Chomsky states, “In the study of any organism or machine, we may distinguish between the abstract investigation of the principles by which it operates and the study of the physical realization of the processes and components postulated in the abstract investigation.”³ Certainly, not all procedures of musical production allow themselves to be expressed formally. The composer’s (or improvising musician’s) mode of operation and level of creativity are neither constant nor continuous, but, allowing simplification, we can outline the borders between three of the sub-processes that are involved: analysis, composition, and diffusion.

PROCESSES IN MUSICAL COMPOSITION

In a first phase, the musician internalises exterior stimuli through interpretation and creates models of various degree of formalisation. In the second phase, composition or “putting-together”, aspects of the material are negotiated through experiments, rehearsals and so on. In the third phase, the internal representation is realised as sound, using an instrument. In poetic terms we may call these phases Listening, Composition and Playing. For an improvising musician the situation is similar. Implementing a model of the three processes in a computer program, we isolate two main parts: first a silent part characterised by analysis, and secondly, a sounding part, characterised by production. Linking the two parts is a transformation function, which is internal and may be invisible to an outside observer.⁴ The nature of such a transformation is evasive in the case of the composition process, and even more obscure in the case of a real-time improvisation performance. Ideally, a computer model to investigate music cognition should be capable of handling large numbers of parameters, of adapting given sets of performance activities to varying situa-

tions, and weighing the value of available strategies. “Leçons” in its present state implements a modest approach to the phenomenon of invention. The computer follows a performance script (see below) containing the actions defining a performance. It does not know any notes or phrases and throughout the performance, it listens to the saxophonist, analyses the audio and builds a database of material which allows it to improve its playing. The computer registers aspects of the style of a human improviser. The material is recombined using a statistical method (Markov chains, generating algorithmically its phrase responses (see below) which are played in real-time. The goal of “Leçons” is to construct a situation where the music is created during the performance and where the two improvisers are compatible in terms of producing musically interesting material.

PROCESSES IN LEÇONS Interactive performance system

I will now describe how the computer part works. The illustration above shows the three main processes: analysis, recomposition, and synthesis. The program is implemented in MaxMSP⁵. The global form, described in the performance script, is a set of variations, and connects back to the master – apprentice idea. Each variation consists of two parts: first, the saxophonist plays solo while the computer listens; then, the two play together.

AUDIO ANALYSIS AND CATEGORISATION

In the Listening part, the computer performs automatic analysis on a continuous audio flow. Required here are methods for data reduction and storage efficiency. The analysis works simultaneously on three different time scales. Going from longer duration to shorter, the data is interpreted as Phrases, Note-categories and Note-models. The high-level musical phenomena are mapped to simple parameters in such a way that they can be manipulated later, in the process of algorithmic composition that generates the computer's phrase responses.

- “Phrases”: an amplitude follower picks up the pattern of the sax player's alternations between playing and silence. The database stores an image of the phrase as a series of windows representing these alterations, which can be stretched and applied to any phrase response.
- “Note-categories”: the audio stream of the saxophone's melodic playing is chopped up into Notes by detecting pitch⁶ and attack (or rather “end-of-silence”). When certain time-related thresholds are passed (these have to be found during rehearsals through trial-and-error) the detection of a new note is triggered. It is defined by its duration, average amplitude and average pitch and these values are approximated in 22 pitch registers, 4 nuances of amplitude, and 7 durations (logarithmic), called “coding classes”. The coding class decide into which of the Note-categories⁷ the new note is stored, or rather, its (non-approximated) pitch-amplitude-duration values (see detail of the user interface below).

SOUND SYNTHESIS

The computer plays an ensemble of four virtual instruments. One takes on a lead role, while the three others are accompanying. The real-life musical metaphor would be that of a group of musicians, something of a cross between a traditional Japanese ensemble and a jazz combo.

- **SashiGulenisu**: the lead virtual player, playing a granular synthesis instrument with a *hichiriki*-like timbre, plays the newly generated phrase and applies the Notemodels to the notes to give it more pitch and amplitude liveliness. There is an automatic algorithm (Brownian motion) that changes the granularity in order to further enrich the timbre.

The notes in the phrase reply are passed on to the accompanying instruments, which derive a notion of harmony by checking all the intervals.¹⁰

The function takes into consideration not only the pitches, but weighs in their duration and amplitude as well (see detail of the patcher below).

- There are two string instruments, both based on Karplus-Strong synthesis¹¹. Here, I wanted some of the sound character of a *Biwa*. The Kastro-Shami has an algorithm for different playing modes: either plucked or in a time-varying tremolo. In either case, it may introduce a glissando between notes. The KastroKoord plays three-note chords.

- The MeksiBass¹² uses algorithms to create something of a jazz style walking-base. Pitches harmonically closer to the fundamental are favoured, in order to strengthen the harmonic unity of the virtual ensemble.

PERFORMANCE SCRIPT

The performance script is the equivalent of a musical score. As an example of what the script looks like, consider the excerpt shown below. The fifth variation has a total duration of 35 seconds (Listening and Playing parts together); all four instruments except MeksiBass will play; the KastroShami will play very slow-moving phrases (3 times longer than the

others) ; the Markov phrase generator will use the most recent forth of the detected notes (the “memory area” is set to 90 120; having access to all the notes would be 0 127).

[5, movementDuration 35 \, instrumentsActivity 1 0 3 1 \, setMemoryArea 90 120]

CONCLUSION

We may now consider the performance from a listener’s perspective. I refer to the sound examples (1: beginning, 2: towards end) taken from a live recording. When setting out, the computer knows nothing and will perform hesitantly. As it gathers more material, the playing is considerably enriched. “Leçons” implements a procedural system, without attaching musical sense to the data it perceives.¹³ When considering the nature of the musical interactivity, the role of the computer is determinant. In an *Automatic* system the machine consists of algorithms for the generation of music, and local unpredictability is often the wanted result. In a *Reactive* system, the computer is treated as an instrument to be performed on, and emphasis is on user interface and direct mapping of controller gesture to sound production. In an *Interactive* system there is a two-way exchange between musician and machine of information that transforms surface elements and deeper structures of a performance in real time¹⁴. The interactivity in “Leçons” is strong in that the composition is decided during the performance, resulting from an automatic analysis of audio. The interactivity is automatic in that it relies on (prefabricated) algorithms to make musical sense. The computer improviser is not a reactive instrument, and the saxophonist needs to invest time to explore the kind of playing which works within the system’s technical limitations and ubiquitous musical setting.

¹ LINDBORG, PM.: *Leçons pour un apprenti sourd-muet*, for two improvisers: soprano saxophonist and computer. (1999, 10’). Composition project as part of the IRCAM Coursus 1998-9. Vincent David, saxophone, Michail Malt, musical assistant. Published by Norwegian Music Information Centre <http://www.mic.no>. The title translates into “music lessons (or “the sound”, *le son*) for a deaf-and-dumb apprentice”.

² LANGENDOEN, T. : “Linguistic Theory”, article 15 in *A Companion to Cognitive Science*. Bechtel and Graham, editors. Blackwell Publishers, Oxford 1999, pg. 235-44.

³ CHOMSKY, N. : *Rules and Representations*. Columbia University Press, New York 1980, Quote from pg. 226.

⁴ MCCAULEY, S. : “Levels of explanation and cognitive architectures”, article 48 in *A Companion to Cognitive Science*, id., pg. 611-24.

⁵ PUCKETTE, M., ZICARELLI, D. : MaxMSP : <http://www.cycling74.com>

⁶ PUCKETTE, M. : “Real-time audio analysis tools for Pd and MSP”, Proceedings of ICMC 1998.

⁷ In this case, since $22 \cdot 7 \cdot 4 = 616$.

⁸ During a 10-minute performance, around 600 notes are picked up, so the number of Note-categories seems reasonable.

⁹ RUSSEL, S. & NORVIG, P. : *Artificial Intelligence, a Modern Approach*. Prentice-Hall International, Inc. Upper Saddle River, NJ. 1995, pg. 34.

¹⁰ Algorithm based on: MURAIL, T : “Esquisse Library” for *OpenMusic*. IRCAM, Paris, 1997.

¹¹ Instrument is based on: SERAFIN, S. : “kastro~” External object for MaxMSP, <http://www-ccrma.stanford.edu/~serafin/> 1999.

¹² Instrument is a wrapper for MEKS, abstraction for MaxMSP. See MSALLAM, R., DELMAR, O., MÅRTENSSON, P. & POLETTI, M.: : “MEKS: Modalys to Extended Karplus-Strong.” IRCAM 1998.

¹³ CARDON, A : *Conscience artificielle et systèmes adaptatifs*. Eyrolles, Paris, 1999, pg. 281-3.

¹⁴ LINDBORG, PM. : *Le dialogue musicien-machine : Aspects de systèmes d’interactivité*. Mémoire de DEA, Université de Paris-4 Sorbonne 2003. <http://www.notam02.no/~perli>