Live Coding the computer as part of a free improvisation orchestra of acoustic instruments

Antonio José Homsi Goulart
Computer Science Department
Institute of Mathematics and Statistics
NuSom – Sonology Research Centre
University of São Paulo – USP
antonio.goulart@usp.br

Miguel Diaz Antar
Music Department
School of Communication and Arts
NuSom – Sonology Research Centre
University of São Paulo – USP
miguel.antar@usp.br

ABSTRACT

In this paper we present our experience of having a live coder amongst acoustic musicians in a free improvisation orchestra. The acoustic musicians in the orchestra had no previous experience with live coding. We will discuss all the difficulties we experienced and the process for overcoming them, illustrating our observations with audio excerpts from some of our recorded sessions. We will also go through a discussion about the nature of the instruments and raise the question of how deeply the musicians should understand code in order to effectively interact in the free improvisation context. Finally we propose a piece for live coder and orchestra that artificially explores the different dimensions of an orchestra session.

1. INTRODUCTION

Both authors of this paper are members of a free improvisation orchestra[1]; Antonio is the live coder and Miguel is the bass player. For 3 years we have been collecting thoughts, sessions records, performances videos, and discussions related to the art of programming audio on the fly amongst other musicians using classical acoustic instruments. In this paper we share these ideas and facts and some of our theories. We discuss some exercises we use in order to develop a good interaction channel between the musicians and we also propose a piece for live coder and orchestra in which we elaborate on a possibility to tackle the challenges in such a scenario.

Albeit sharing some concerns like participation, communication within the orchestra and with the audience, our approach is different from those of other works relating improvisation and laptop ensembles or laptops in ensembles (Burns 2012, Lee and Freeman 2014, Ogborn 2014). Although some of these groups incorporate acoustic instruments occasionally, our orchestra is not based on laptops, but only features one live coder joining instrumentalists playing alto and baritone saxophones, trombone, didgeridoo, flute + effects, double bass, prepared piano, and voice + everyday objects.

Wilson et al. (2014, 2) talk about the role of "intervention mechanisms to facilitate aspects of musical interaction that can be difficult to realize in strictly free performances". They use networked music systems for the exchange of information in a context of a laptop orchestra. But in our case we are using live coding as a stand-alone instrument, and we are concerned about how the orchestra musicians can interact better, understanding the live coder issues and possibilities. We think that by sharing what we have observed and experimented with we can help and also hear back from other groups working in a similar context.

[1]Orquestra Errante (OE) is an experimental group based on the Music Department of the School of Communication and Arts at University of São Paulo, founded in 2009 as a laboratory to the practice of free improvisation and reflections about its theories and territories. OE membership is seasonal and variant, usually composed by students and professors from the Music Department, with some sporadic students from other departments. OE is also open to occasionally hosting visitor musicians from outside the academy. In our weekly meetings we practice free improvisation sessions and also oriented exercises, with a round of discussions after each piece. We keep crude records of our improvisations so our members can access them for later reasoning and reflections for their research. Some samples of our work, which we refer to later in this paper, can be found at *soundcloud.com/orquestraerrante*

OE members' background comes from classical music, and some also studied jazz. They have been through a lot of instrument practice and conventional ear training, but only recently are also starting to explore audio effects and electronics as part of their set of tools, and getting used to electro-acoustic / electronic / computer music repertoire and possibilities. In the first year of OE with live coding a common fact that we were observing in our sessions was that most times the live coded sound acted as a set of background layers with little influence on the acoustic musicians, while the live coder could not find a way to interact either. However, by the time the group understood better the nature and range of live coding, and the live coder got better at his activity, the group started to evolve towards a better interaction and wholeness.

In the remainder of the article we will go through some thinking about free improvisation in Section 2, the differences about the instruments involved in our group and their manipulation in Section 3, and then in Section 4 we present the interaction issues we went through in our sessions. In Section 5 we describe an interesting idea for a performance we developed, before concluding the text.

2. FREE IMPROVISATION (FI)

As a musical performance FI in a group presupposes an intentional collective and collaborative activity. Involved musicians must, through sound, start a conversation which evolves in an uninhibited manner, and each performer intervention simultaneously creates, modifies and draws the way (Costa 2003, 42)[2]; as in a spontaneous conversation, the subjects and their unfoldings are diverse. This metaphor highlights one of the major characteristics of FI, that of the creator / interpreter role of each participant. Within an FI session the unraveling of a performance is dictated merely by the interactions between them, in a sound flow shaped along the *devir* of the act. Free improvisers play a musical ludic game not expecting the victory, but individually proposing ideas that may or may not be corroborated by the collective (Falleiros 2012, 59)[3]. Costa (2007, 143) argues that "There is a presupposition that everything is impermanent and the forms are provisional aspects of intermediations made feasible by unexpected and rhizomatic connections".

Another aspect of FI is the deliberate act of trying to overcome established musical idioms in search of a "non-idiomatic" improvisation (Bailey 1993, xii), and an openness to the possibility of using any sound in the discourse. In such a way free improvisers are always looking for different ways and techniques to extend their instruments, many times during the course of a performance. In such a way we can neither know nor expect anything from a FI session. There is the potential for absolutely anything regarding sonorities and interactions. It is a collaborative (when in a group) experimentation process, radically striving towards real time music creation. Usually there is no previous indication about the musical content, but FI is also open to minimal scripts with any type of instructions that can be revealed in advance or when the session is about to start.

Previous information about the other performers is not a prerequisite for 2 or more people to free improvise together. Musicians should be able to interact based only on the sound they create and hear, not needing to know anything about each other's instruments' sounds and way of playing (we address that problem later). For instance, the bass player should be able to jam with the drummer even not knowing anything about drum sticks or the relation between the different drums and their timbres. In such a way an active hearing is a crucial aspect in FI; Costa (2007, 143) argues about the necessity of a Schaefferian reduced listening, abstracting "the sound sources, the musical meanings and the inevitable idiomatic gestures present in each musician's enunciations". In other words, when free improvising we should focus our attention to the sonorous aspects of the instrumental actions.

Falleiros argues that along with this active listening, a promptness for a benevolent interaction is also a must, where "(...) improvisers are distant from manipulative or polarizing actions, (...) which drive the listening in a previous direction that presents an established way to deal with". This benevolent intention resembles a chamber music orchestra attitude, in which "the first step to learn to correctly execute chamber music is learning not to exhibit oneself, but to recede. The whole is not constituted by an auto-affirmation of the individual voices" (Adorno 2009, 190, our translation). The same goes for FI, where this chamberistic posture (courtoisie[4]) permeates through the intermediation of the sound fragments and enunciations.

[2]All translations from Costa were made by us from the original in Portuguese.
[3]All translations from Falleiros were made by us from the original in Portuguese.

3. MANIPULATING MUSICAL INSTRUMENTS

Some theories characterize the musical instrument as an extension of the body (Ihde 1979; ___ 1980 apud Magnusson 2009, 1), assigning to the musician the role of improving this condition up to the point of being able to fluently express musical intentions, or as put by Nijs (2009, 2), "(...) the musician no longer experiences a boundary between himself and the instrument". In the context of improvised music this ability is crucial, as usually there is no previous structure or composition to guide the interaction between the musicians and guarantee coherence in the music created, which depend then on the performers' sensibility and dexterity.

The body is fundamental in a music based on sounds and empirical gestures on the instruments. The FI process is radically related to body and gesture, leaving no room for separation between body and mind (Costa 2003, 94-95), so there is an inherent physicality in the musical practice which varies according to the musician and the instrument, but regardless of both, conveys a sensation of immediacy from physical stimulus to sound response. So, in other words, there is a body responsible for creating immediate sounds using an acoustic instrument as an extension of the body.

Programming the computer on-the-fly, however, the relationship between musician and instrument changes from a physical-mechanical (body-instrument) to a physical-abstract (body-code) one. Notice that although we agree with Magnusson (2009, 1) when he interprets the computer instrument as a mind extension, we still put it as an extension of the body, in part because the musician should write code, an action which takes time on stage, and also because in an orchestra context the body is still there as part of the group and as visual information to the other performers and audience.

Compared to acoustic instruments, it takes more time to realize a musical intention in an algorithm (which then outputs sound). This lag comes both from the time needed to elaborate the algorithm and the actual writing (sometimes done together), and regardless how fast a performer is in typing and how literate about the techniques (Nilson 2007, 4), unless snippets with basic blocks and specific sounds are used, the implementation of a musical intention is not immediate. However, albeit imposing immediacy restrictions on the performance, live coding enables the performer to implement gestures for the future. As pointed out by Stowell and McLean (2013, 3), "a live coder, like any improvising performer, is under pressure to do something interesting in the moment. Live coders can use abstraction and scheduling so the notion of 'in the moment' may be a little different to that for more traditional instruments improvisers".

Besides that there are the vast possibilities offered by live coding, compared to acoustic instruments or fixed graphical user interface based computer instruments (Blackwell and Collins 2005, 6). Theoretically any sound can be generated with the different synthesis techniques and audio effects, and the musical decisions are only limited by the coder ability and creativity (and also the pressure due to time lags).

Finally there is the projection issue. TOPLAP manifesto[5] asks to "show your screens", but projecting the code to the audience is not unanimously seen as a good choice (McLean et al. 2010, 1). In our case the live coder is part of an orchestra, so at first we decided not to project because we thought that it would bring most of the attention to the live coder and his code. Our usual audience is more used to acoustic instruments and as a result of this the projection would have a boosted shock factor (Collins 2011, 3) which could in turn blur their attention to the group music and performance. But then we tried the projection at one of our practice meetings, and it makes more sense to talk about that in the next section.

4. INTERACTION IN PRACTICE AND SOME OF OUR EXERCISES

The same way that the bass player and the drummer can jam without staring each other, should they be able to jam with the live coder without reading his code? Does reading his future processes contribute to the improvisation, or does it give too much information in advance?

---

[4]French word for courtesy, politeness, civility, which appears in Adorno's text two paragraphs before the excerpt we cited.

[5]http://toplap.org/wiki/ManifestoDraft

We observed that in our group the complex relations established at the FI process are made of the physical, corporal and emotional dimensions from the instrument gestures, and the musicians' attention fluctuates between the sonorous and the visual.

We understand that the reunion of a live coder with musicians playing acoustic instruments offers a rich context for the creation of very interesting improvised music, being up to everyone to understand everyone's potentials and limitations. Regarding live coding, as put by Wilson et al. (2014, 1), "On the one hand, it becomes possible to do many previously unimaginable things; on the other, it is often not possible to do them very quickly".

At first OE musicians did not seem to understand these lag issues regarding programming audio on-the-fly. The tendency in our performances was to observe conversations between the acoustic instruments, while the live coded sound remained as a background layer without much interaction, most times only influencing and being influenced by the orchestra in the long term. An example of this background layer can be observed from 00:58 up to 02:30 in the piece[6] "Estados, eventos e transformações", where the high-pitched sounds from live coding were not influencing the acoustic musicians. However, the same sound pattern explored with a faster tempo drives at 02:35 the recovery from a sort of break and triggers a beautiful movement from the piano player at 02:55.

One of the tools we use to articulate the flow is to explore moments of duos or trios, either in a spontaneous fashion or previously settling the groups and when / how they should take place. For instance, we always warm up for a session making a circle and practicing interactions in duos. Two musicians start jamming, when appropriated the next one on the circle enters in the interaction and they improvise as a trio for a while, then the one who was in the trio for most time leaves, the remaining duo improvise for a while, and so it goes until everyone jammed. Each duo would typically last for about 2 to 3 minutes. Another problem we observed was the extreme difficulty for the live coder to interact in these short moments of duos. This difficulty can be observed in the example "Transição de duos – Set2014", where duos transitions were explored. Notice that good interactions happen up to 05:00, when starts the live coder and flautist turn and they struggle to contribute and support each other. The only attempt to interact was based on imitating attacks up to 06:16 (interesting musical output, though), when the live coder recedes in order to prepare something for the wrap up of the exercise and the flute continues. Lack of time might be pointed as the main obstacle here, forcing a tendency of using snippets or coding simple percussion or melody lines with the basic waveforms. But it also seems to exist in almost everyone a kind of anxiety to make a remarkable movement in the current turn, and we think that within this scenario the discourses are unlikely to converge.

Although we eventually made good performances[7], it was clear to us that more information about the live coding process and computer music techniques were lacking for OE members. Up to that point our coder was using SuperCollider (mainly the Patterns library (Kuivila 2011) and some simple one-liners), so we talked about some synthesis techniques, how long takes in average to code them as SynthDefs, what can be done with the Patterns after the synthesis definition is coded, and how huge turnarounds in output can be obtained with small twists in code. Finally, we made our practice session projecting the code (half the orchestra have not seen live coding before), and they got a better idea about the live coder different way of playing, like, as beautifully put by McLean (2008), "In contrast to musical instruments directly plucked, struck and bowed by a human performer, the movements of a live coded performance all happen inside a computer. The live coder may be typing furiously, but it is the movement of control flow and data inside the computer that creates the resonances, dissonances and structure of music".

In the session "at MusicaNova 2014", which we considered overall successful, a duo between the live coder and the trombone player starts at 04:20. The first one struggles to make something interesting, and while only a slow tempo rhythmic tone is his actual sound, the latter makes his best with a nice sewing around the live coder poor sound, carrying the duo for a while. By the time the live coder finally fixes the code and get what he was trying to, the trombone player is immediately influenced and a nice interaction and new sonority are obtained by the duo, which then decides to continue exploring it until 07:20, when the bass

[6]All pieces that we use as example can be found at *soundcloud.com/orquestraerrante*

[7]The second half of the piece "Buscando chaves" is in our opinion an example of good interaction between all the performers (everyone active listening everyone's musical ideas and supporting each other) and interesting musical result (contrasts, nuances, resolutions of tensioning, *crescendos* and *diminuendos*).

player comes in. Another interesting moment in this piece (from the live coding point of view) happens from 25:40, with a buzz that morphs into impulses, which increases in density and drives a nice conversation between all instrumentalists. This is then followed by a sort of break with the coded clicks and acoustic textures in *piano*. At 27:35 there is a huge drop in the coded sound where only a regular click is left, and the musicians unanimously stop for a while and then come back for another *tutti*, showing perfect interaction in this long sequence of tension and release. Also in the performance "at MusicaNova 2014", there is a very intense *tutti* passage at 31:06 where the live coder does achieve interaction, influencing and being influenced by the acoustic musicians' quick movements and textures transitions.

And now we come back to the questions posted in the beginning of this section, with another question. OE members do not know programming, and even if they knew, how feasible would it be to simultaneously perform their instruments, explore extended techniques, active listen to everything, and also read code?

We adapt the free improvisation presupposition and state that it is possible to interact only through sound, given that everyone have and idea about everyone's instrument nature. People playing acoustic instruments should know that in live coding (usually) there is not a direct relationship between typing and sound output, and writing sentences (the live coder gesture) - takes time. With that, the acoustic musicians can support the live coder lots of different ways while an algorithm is being written or bugs being fixed. Instrumentalists should also keep in mind that once the initial algorithm is written, with little time and typing lots of variations and musical intentions can be achieved and explored collaboratively.

However, if the musicians (except the live coder) in the orchestra are able to read code and understand where the live coder is sonorously going, interaction with the future is enabled and the musicians can also experiment making room (or creating tension) for the coded sounds that are to come (the resolution). If only one or a few of the members can read code they can still make this bridge to the future (and if their musical movements are convincing they can lead the way in that bridge), but the pitfall in this case would be an unconscious process of paying too much attention to this one aspect, "overlistening" to the other members, ruining the lack of differentiation between the musicians in a free improvisation scenario. Maybe (?) an interesting scenario for an acoustic orchestra with a live coder would be if everyone could read code and interact with the future, also being able to, preferably via sounds, warn the live coder about tortuous ways.

Of course not every free improvisation orchestra with acoustic instruments members might want to learn programming and digital signal processing algorithms. But the same way that the live coder should know something about his fellow musicians' possibilities and limitations, like a composer when choosing specific instruments for writing pieces, it is extremely important that all orchestra members are at least exposed to the process of live coding, so they can be aware of its specificities.

5. DEVIR (FOR LIVE CODER AND GHOST ORCHESTRA)

In this piece we wanted to explore the relations between the space where the improvisation happens, the visual exchanges between the musicians, the interactions based on the immersion on the sounds and the different possibilities of a musical discourse with a specific material.

As we observed and mentioned before, the sonorous is often blurred by the visual in collective musical practices. We also discussed about the live coder's different types of gestures and peculiarities that the group should take into account when interacting. So we wanted to artificially create a context in which everyone is forced to interact with the live coder, actively listen to the live coded sound, and where the live coder actually tries to suggest how the individual aspects of the resultant orchestra music will be imagined.

In Devir, the musicians are divided in duos which are placed acoustically and visually isolated. The live coder and audience are placed in yet another room, where the code is projected for the audience and the live coder controls a mixing board into which a mono signal from each duo and the stereo signal from the computer are connected. The live coded audio is also sent to one musician from each duo through headphones.

In the performance, the live coder controls the volume of each duo and the live coded sound in the final mix, the one that is reproduced in the room with the programmer and audience. The live coder also controls the volume of the computer sound that one musician from each duo will receive in the headphones. So the live

coder is faced with the challenge of coming up with a live coded sound that at the same time is interesting for the final mix and suggests directions for one musician in each duo, which then interacts with the coded sound, guiding the interaction within their partners within the duos.

In some moments the live coder might find it interesting to use only the computer sound in the final mix, or use it in conjunction with only one duo (for instance to emulate a trio even when the full orchestra is playing) or a few of the duos sounds (to artificially leave out of the performance a part of the orchestra for a while). Another interesting possibility is using only the duos sounds in the final mix, individually suggesting directions for each duo (voiding the volume of the headphones for all but one duo at a time).

Interestingly, in this piece, the live coder role is actually to live code both his algorithm, the entire orchestra, and also the final mix and musical result of the session. Of course, depending on the technical structure of the venue some variations for what was proposed can be tested, for instance isolating all the musicians and individually communicating with them, or also to try a scenario with more than one live coder, perhaps even one for each acoustic musician, and the live coders talk to themselves like a clan playing first-person shooter video games[8] (either shouting in the same room or with headphone sets over the network).

Aside from the complexity regarding the necessary hardware, we don't suggest projecting the code for the duos because we want the musicians to know that the computer musician is live coding but we do not want them to have to be able to read the code, as in a regular setup they wont have time for it, or many would not even be able to, like we considered earlier. But occasionally, if conditions permits and some musicians can read code, another variation would be to project the code to one musician in each duo and send audio over headphones to the other one, creating another dimension within each duo. Another more extreme variation would be to isolate everyone and send audio to half of them and projection to the other half, or even only project the code to all isolated musicians and mix their individual sounds of interaction with the *solfeged* code.

6. CONCLUSIONS

In this paper we considered the collision of two extreme musical practices in an acoustic orchestra context. In free improvisation musicians' enunciations do not go through any kind of filter. This is an intentional liberation from music traditions. The only limits regarding sonorities and musical conversations are the performers' creativity and dexterity.

Above all the difficulty involved in finding a definition for live coding, there is an agreement that it "(...) involves the writing of algorithms in real time" or, concatenating another two excerpts by Magnusson (2014), is a "method of composing through performance in real time", involving "deep exploration of the medium at hand". It is striking how this last description is intimately related to broader definitions of free improvisation (Falleiros 2012, 18). And we agree with Wilson et al. (2014, 1) when they say that "live coding might be seen as an improvisational interface par excellence but lacks in agility".

Given all the difficulties we went through and the differences between acoustic instruments and code as an instrument, we were at first wondering if live coding was adequate as part of a primarily acoustic FI orchestra. We think it is, but important concepts must be worked by all the members to enable fluency in conversations within the flow of a session. It is desirable that the acoustic musicians new to live coding attend some performances or watch good videos, preferably accompanied by the orchestra coder. They should have an idea about the processes of writing, debugging, adapting and reusing code so they can tell what is going on in interaction situations when improvising with the live coder.

We also considered the necessity of coding/DSP knowledge by the acoustic musicians. Of course that it would enable them to come up with interesting anticipations while improvising, but we do not think it is a need, and we should not forget that FI presupposes interaction based solely on the sound up to the current moment.

Up to this point we still have not experimented with incorporating in the live coding possible rhythmic information that might arise in an improvisation. Although rhythm patterns are usually avoided or

[8]http://en.wikipedia.org/wiki/First-person_shooter

disguised in FI practices, it could be interesting to deal with rhythm synchronization issues between the acoustic and computer sounds.

The possibility of interacting with the future would not be a stray from Bailey's (1990, 95) "playing without memory" paradigm. But do we want them to read code? Agreeing with McLean and Wiggins (2010, 2) when they talk about the code written by live coders, "(...) Their code is not their work, but a high level description of how to make their work". In free improvisation there is a priority relationship with sound, and not respecting that would be going too far from the free improvisation paradigm.

No, we do not want them to need to read code. And programs sometimes crash. But that is not a problem if we have programmers playing acoustic instruments in Orquestra Errante. The interaction between all the performers should be based on the sounds at the current instant allied with a benevolent attitude and an openness to the future, in an expectation projected through code *solfege*, not imprisoned by its abstract interpretation, but derived from its sonorous concreteness.

Acknowledgments

We would like to thank all members of Orquestra Errante for all the practices and discussions over the last 3 years. Our work was partially funded by NuSom (Sonology Research Center at USP) and CAPES (proc. Number 8868-14-0).

REFERENCES

Adorno, Theodor. 2009. *Introdução à sociologia da música*. São Paulo: Fundação Editora da UNESP.

Bailey, Derek. 1992. *Improvisation: Its nature and practice in music*. England: Moorland Pub.

Blackwell, Alan and Nick Collins. 2005. "The programming language as a musical instrument". In: Proceedings of the PPIG05 (Psychology of Programming Interest Group).

Burns, Christopher. 2012. "Adapting electroacoustic repertoire for laptop ensemble". In: Proceedings of the 1st Symposium on Laptop Ensembles and Orchestras, 76-79.

Collins, Nick. 2011. "Live coding of consequence". Leonardo 44 (3): 207-211.

Costa, Rogério. 2003. *O músico enquanto meio e os territórios da Livre Improvisação*. PhD thesis, Pontifícia Universidade Católica, São Paulo.

_____. 2007. "Livre Improvisação e pensamento musical em ação: novas perspectivas". In: Notas Atos Gestos, edited by Silvio Ferraz, 143-178, Rio de Janeiro.

Falleiros, Manuel. 2012. *Palavras sem discurso: Estratégias criativas na Livre Improvisação*. PhD thesis, Universidade de São Paulo, São Paulo.

Kuivila, Ron. 2011. "Events and patterns". In: The SuperCollider Book, edited by Scott Wilson, David Cottle and Nick Collins, 179-206. MA: MIT Press.

Lee, Snag Wong and Jason Freeman. 2014. "Real-time music notation in mixed laptop-acoustic ensembles". Computer Music Journal, 37 (4): 24-36.

Magnusson, Thor. 2009. "Of epistemic tools: musical instruments as cognitive extensions". Organised Sound 14 (2): 168-176.

_____. 2014. "Herding cats: Observing live coding in the wild". Computer Music Jornal 38 (1): 8-16.

McLean, Alex. 2008 . "Live coding for free". In: FLOSS+Art, edited by A. Mansoux and M. de Valk, 224-231. London: Open Mute.

McLean, Alex and Geraint Wiggins. 2010. "Live coding towards computational creativity". In: Proceedings of the International Conference on Computational Creativity, 175-179.

Mclean, Alex, Dave Griffiths, Nick Collins and Geraint Wiggins. 2010. "Visualisation of live code". In: Proceedings of the EVA'10 (2010 International Conference on Electronic Visualization and the Arts), 26-30.

Nijs, Lunc, Micheline Lesaffre and Marc Leman. 2009. “The musical instrument as a natural extension of the musician”. In: Proceedings of the 5th Conference on Interdisciplinary Musicology.

Nilson, Click. 2007. “Live coding practice”. In: Proceedings of NIME'07 (7th International Conference on New Interfaces for Musical Expression), 112-117.

Ogborn, David. 2014. “Live coding in a scalable, participatory laptop orchestra”. Computer Music Journal, 38 (1): 17-30.

Stowell, Dan and Alex McLean. 2013. “Live Music-Making: A Rich Open Task Requires a Rich Open Interface”. In: Music and Human-Computer Interaction, edited by Simon Holland, Katie Wilkie, Paul Mulholland and Allan Seago, 139-152. London: Springer.

Wilson, Scott, Norah Lorway, Rosalyn Coull, Konstantinos Vasilakos, and Tim Moyers. 2014. “Free as in BEER: Some explorations into structured improvisation using networked live-coding systems”. Computer Music Journal, 38 (1): 54-64.