INTERACTIVE COMPOSITION AND IMPROVISATION ON THE HYPER-FLUTE

Cléo Palacio-Quintin
Faculté de musique, Université de Montréal
IDMIL – CIRMMT, McGill University

Mark Zadel
IDMIL – CIRMMT
McGill University, Montreal

ABSTRACT

This paper briefly presents the original design of the hyper-flute and then explores interactive composition strategies for this augmented instrument. Design approaches of a real-time performance environment devoted to musical improvisation are discussed.

1. INTRODUCTION

Since 1999, I have been performing on the hyper-flute [11]. Interfaced to a computer via electronic sensors, the extended flute enables the direct control of various digital processing parameters that affect the flute's sound while performing and allows the composition of unusual electroacoustic soundscapes.

My interactive composition strategies for the hyper-flute have been influenced largely by my practice of improvised music. I will present my ideas on the development of musical structures in relation to a computer-based improvisation environment. Perspectives on the development of the hyper-flute and of a hyper-bass-flute are also addressed.

2. THE HYPER-FLUTE

By the end of my studies in contemporary flute performance (Université de Montréal, 1997), I was heavily involved in improvised music and had started looking for new sonorities for the flute in my own compositions. Already familiar with electroacoustic music and with the use of the computer, it was an obvious step to get into playing flute with live electronics. My goal was to keep the acoustic richness of the flute and my way of playing it. The computer would then become a virtual extension of the instrument.

During post-graduate studies in Amsterdam, I had the chance to meet the experienced instrument designer Bert Bongers [2] and the *meta-trumpeter* Jonathan Impett [7] at the Dartington International Summer School of Music (U.K.). Several months later, I registered as a student at the Institute of Sonology in The Hague (The Netherlands) in order to build my hyper-flute. The prototype of the hyper-flute was mainly built during the fall of 1999 with the help of Lex van den Broek. Bert Bongers was a valuable consultant for the design and he also made the main connector from the sensors to the Microlab interface.

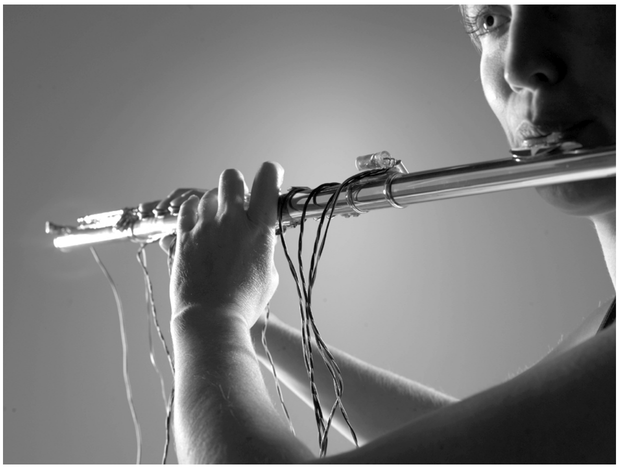

Figure 1. The hyper-flute played by Cléo Palacio-Quintin. Photograph by Carl Valiquet.

2.1. Original Design: Interface & Sensors

The interface used with the hyper-flute is a Microlab, originally designed and developed by J. Scherpenisse and A.J. van den Broek at the Institute of Sonology. This electronic interface converts the voltage variations from various analog sensors into standard MIDI data. It offers 32 analog inputs, a keyboard matrix of 16 keys and an integrated ultrasonic distance measuring device.

Inspired by Jonathan Impett's meta-trumpet, I put different types of electronic sensors on my flute. "As far as possible, this is implemented without compromising the richness of the instrument and its technique, or adding extraneous techniques for the performer — most of the actions already form part of conventional performance." [7]

There is little free space to put hardware on a flute because of the complexity and small size of its key mechanism. Nevertheless, it was possible to install sensors at specific strategic locations. Table 1 shows an overview of the sensors originally installed on the hyper-flute.

Several analog sensors send continuous voltage variations to the Microlab which converts them into MIDI Continuous Controller messages. Ultrasound transducers are used to track the distance of the flute from the computer. Pressure sensors (Force Sensing Resistors) are installed on the principal holding points of the flute (under the left hand and the two thumbs). Two magnetic field sensors (Hall Effect) give the exact position of the G# and low C#

Table 1. Sensors installed on the hyper-flute

Sensors	Parameter
1 Ultrasound sensors	flute's distance to computer
3 Pressure sensors (FSRs)	pressure: left hand and thumbs
2 Magnetic field sensors	motion of G# and low C# keys
1 Light-dependent resistor	ambient light
2 Mercury tilt switches	tilt and rotation of the flute
6 Button switches	discrete cues

keys, both operated by the little fingers. A photoresistor that detects the variations of ambient light is positioned on the headjoint of the flute.

Other controllers used on the hyper-flute send discrete values (MIDI note on/off messages). Two mercury tilt switches are activated by the inclination (moving the footjoint up) and the rotation (turning the headjoint outwards) of the instrument. There are also six small button switches. Two of them are located on the headjoint, and two are placed close to each of the thumbs and can be reached while playing.

Performing with some of the sensors installed on the hyper-flute was not always compatible with standard flute technique and entailed a long learning process. Experience has shown how much the interaction between acoustic playing techniques and the motion captured by the sensors is intimately connected. Musical gestures need to be thought of as a whole. Just like learning an acoustic instrument, it is necessary to play on an electroacoustic interface for a long period of time before achieving a natural control of the sound. As on any musical instrument, expressivity is directly linked to virtuosity [5].

3. INTERACTIVE COMPOSING & MUSICAL IMPROVISATION

3.1. Interactive Composing

Joel Chadabe is one of the pioneers of real-time computer music systems. He named this new method of composition *interactive composing*, that he defined in [4]: "An interactive composing system operates as an intelligent instrument — intelligent in the sense that it responds to a performer in a complex, not entirely predictable way, adding information to what a performer specifies and providing cues to the performer for further actions. The performer, in other words, shares control of the music with information that is automatically generated by the computer, and that information contains unpredictable elements to which the performer reacts while performing. The computer responds to the performer and the performer reacts to the computer, and the music takes its form through that mutually influential, interactive relationship."

From this point of view, the performer also becomes an improviser, structuring his way of playing according to what he hears and feels while interacting with the computer. Like his instrument's sound, the performer's role as been extended. In [8], Jonathan Impett also considers that the use of computers to create real-time music redefines the traditional subdivisions in musical practice. "In such a mode of production, the subdivisions of conventional music are folded together: composer, composition, performer, performance, instrument and environment. Subject becomes object, material becomes process."

In most of the cases, users of interactive computer systems in live performance are at once composer, performer and improviser. Due to the novelty of the technology, few experimental hyper-instruments have been built, mostly by musicians who play it themselves. It is quite difficult to define the line between composer and performer while using such an interactive system. The majority of performers using such instruments are concerned with improvisation as a means of making musical expression as free as possible.

3.2. Developing Musical Structures

Using an interactive computer system linked to an augmented instrument, the performer has to develop a relationship with different types of electroacoustic sound objects and musical structures. These relationships correspond to the fundamentals of musical interaction. The computer part can be supportive, accompanying, antagonistic, alienated, contrasting, responsorial, developmental, extended, etc.

All the musical structures included in a mixed piece have different roles. Some affect the micro-structure of a musical performance, others affect the macro-structure of the piece, and many are situated somewhere in between. The interaction between the performer and musical structures vary. The structures can also have different levels of interactivity between themselves. We could divide them in 3 basic distinct types:

- The original acoustic sound is modified by live processing, controlled through the gestural interface. For example, the computer is modifying and/or extending the acoustic sound itself, by routing it through filters, harmonizers or delays. The computer is used as a direct extension of the performer's acoustic instrument.

- Sound is synthesized in real-time using the various interface inputs (gesture information and sound analysis) to control different parameters. Synthesis can respond to the performer's gestures without being directly linked to the acoustic sound of the instrument. Control and sound data can also be recorded and used later during the piece, permitting time stretching and compression.

- An independent sound-track can accompany the flute or play by itself over the course of the piece. It can be pre-recorded, or generated in real-time with the use of computer algorithms. This type of structure is completely independent from the performer's actions.

To create an interactive composition, the composer can include many different sound processing structures, including different levels of hybridization of the three types described. Each of them can also have a different level of controllability and indeterminacy. Different types of random processes can be used for the creation of synthesized sounds, control of algorithms, and control of sound processing parameters. The performer can be asked to control a parameter directly, or its level of indeterminacy. The incorporation of randomness in the composition will generate unpredictable elements, giving the performer the opportunity to really interact with the machine.

A composed piece with a finite form follows the same sequence of events for each different performance. In this context, the computer generates a kind of complex interactive *tape part* that *follows* the performer. However, to play improvised music, the interactive computer environment needs to be designed to maximize flexibility in performance. The environment must give the opportunity to generate, layer and route musical material within a flexible form.

3.3. Musical Improvisation

The term *improvised music* can refer to various musical practices. As discussed by Georges Lewis [9], two important models are *open improvisation*, as practiced by members of the Association for the Advancement of Creative Musicians (an African-American musicians' collective founded in 1965 in Chicago), and *free improvisation*, as practiced by European improvisers, such as Joëlle Léandre, Derek Bailey and Evan Parker. My improvisation practice is situated somewhere between these two models, and also refers to *musique actuelle* as it has developed in Québec since the 1980s [12]. However, my discussion here is concerned with musical improvisation in wide acceptance, as defined by Lewis: "Musical improvisation is . . . an interaction within a multi-dimensional environment, where structure and meaning arise from the analysis, generation, manipulation and transformation of sonic symbols." This definition is especially relevant when using an interactive computer system to perform improvised music, as the computer is an ideal tool to analyze, generate, manipulate and transform sounds.

Since the late 1970s, the performer/composer Larry Ochs has developed strategies for structured improvisation with the Rova saxophone quartet. In designing my interactive system for live performance, I have been inspired by his ideas about how to structure improvisation. "Formal devices/structures are employed to get at the musical requirements of a given piece. It is always the primary goal in any piece to be musically coherent; to tell a story and/or to create a mood, feeling, or environment. The devices used in any given piece are employed with the sole intent of realizing the intentions of that composition. And the decision to use (structured) improvisation as a means of realizing even more — more than the composer imagined possible when composing the piece (or section of the piece). Or, at the very least, to allow for the possibility of different — or fresh — realizations of that intention with each performance." [10] The strategy of composing a piece containing improvised segments has been relevant to the design of my computer environment for improvisation.

3.4. Computer Environment for Improvisation

While programming the computer environment for improvisation (in Max-MSP, in this case), my approach is to compose pieces with a flexible structure, a kind of open form composition. I must first consider which type of musical structures I want to incorporate in my improvisation, and then decide how I want to control them.

Performing improvised music on the hyper-flute, I have focused on the development of the first type of musical structure mentioned previously: directly transforming the flute sound with live digital processing. However, when looking for new extended flute sonorities, the process has also led me to the integration of sound synthesis. I rarely use pre-recorded material; it does not seem appropriate to me for *free improvisation*, as it needs to be predetermined.

Each performer has his own repertoire of different instrumental sounds and playing techniques from which choices can be freely made while improvising. The *sound palette* is very wide, and switching from one type of sound to another is done within milliseconds. Ideally, the computer environment would give the same improvisational freedom that the performer has developed with his acoustic instrument. My goal is to create a *sound processing palette* as rich and complex as the instrumental one. I wish to improvise freely and be able to trigger many different computer processes at anytime, without disturbing my flute playing.

I have developed a modular system combining different Max-MSP patches that can be accessed anytime by using the hyper-flute as the controller. Different transformations of the flute's sound are made by standard sound processing effects like delays, granular synthesis and harmonization. All the parameters of the sound processing can be controlled in real-time by the performer in many different ways. According to Hunt's research on the subject [6], complex mapping strategies have been developed to operate various multi-parametric gestural controls. Most of my processing patches also include some random algorithms that the performer can turn on to let the computer make some decisions. The desired surprise effect creates a real human-machine interaction.

However, digital sound processing patches can only generate sounds that have been programmed (even if they include some random processing) and any interactive gestural interface has a limited number of controllers. The freedom of the performer is nevertheless limited by the computer's environment. In reality, my computer environment consists of a flexible balancing act of composed structures within an open form.

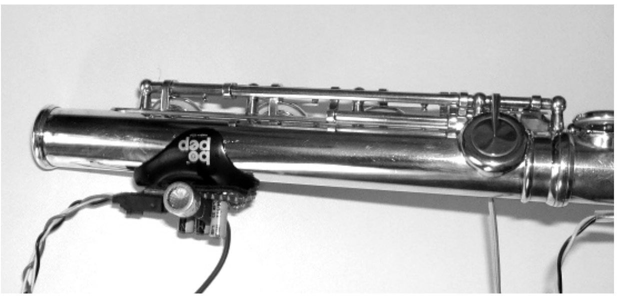

Figure 2. The accelerometer and ultra-sound transducer mounted on a Bo-Pep for the new hyper-flute.

4. NEW PERSPECTIVES

After eight years, I am now very comfortable playing the hyper-flute. I have developed a very good knowledge of my musical needs to control the live electronics while performing. On the other hand, the performance skills and mapping strategies learned over time suggest new directions for the instrument and computer environment.

Though there are always programming issues to address before achieving an ideal interactive computer environment, I have always felt limited by the number of controllers on the hyper-flute. That is what has led me to new developments on the instrument itself. The hyper-flute is in the process of being rebuilt with extra sensors and other enhancements. A two axis accelerometer is placed on the foot-joint of the instrument together with the ultrasound transducer (see Figure 2), and several buttons are added.

A hyper-bass-flute is also in development. The bass-flute has the advantage of being a much bigger instrument, so there is more space to attach hardware. Nevertheless, the weight of the instrument limits the movements of the thumbs to reach different sensors while playing as I do on the hyper-flute, so the design of the sensors needs to be different. Composition strategies will need to be adapted for this instrument, and a new learning period will be necessary to perform with it.

For both hyper-flutes, the Microlab is replaced by a new interface using the Open Sound Control protocol [1]. This protocol will give me the opportunity to use different types of data, with more resolution and bandwidth than the previously used MIDI.

Until now, I mostly used the hyper-flute to perform improvised music. Wishing to expand the repertoire for the hyper-flute, I began doctoral studies in January 2007 to work on written compositions. So, in addition to the development of my improvisational environment, I am now composing written works, and hope to have other composers do so as well.

The new prototypes of hyper-flutes will be easier to reproduce. Eventually that will make it possible for other flutists to perform those new works by transforming their own instrument. As the interest for mixed music is growing, the musical perspectives are very rich for new augmented instruments, like the hyper-flutes.

5. REFERENCES

[1] *Introduction to Open Sound Control (OSC).* http://opensoundcontrol.org/introduction-osc.

[2] B. Bongers. Physical interfaces in the electronic arts. Interaction theory and interfacing techniques for real-time performance. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*. IRCAM - Centre Pompidou, Paris, 2000.

[3] M. Burtner. The metasaxophone: concept, implementation, and mapping strategies for a new computer music instrument. *Organised Sound*, 7:201–213, 2002.

[4] J. Chadabe. Interactive composing: An overview. In C. Roads, editor, *The Music Machine: selected readings from Computer Music Journal*, pages 143–148. MIT Press, Cambridge-London, 1989.

[5] C. Dobrian and D. Koppelman. The ‘E’ in NIME: Musical expression with new computer interfaces. In *Proceedings of the International Conference on New Interfaces for Musical Expression, NIME-06*, pages 277–282, Paris, 2006.

[6] A. Hunt and R. Kirk. Mapping strategies for musical performance. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*. IRCAM - Centre Pompidou, Paris, 2000.

[7] J. Impett. A meta-trumpet(er). In *Proceedings of the International Computer Music Conference*, pages 147–149, San Francisco, 1994. International Computer Music Association.

[8] J. Impett. The identification and transposition of authentic instruments: Musical practice and technology. *Leonardo Music Journal*, 8:21–26, 1998.

[9] G. E. Lewis. Interacting with latter-day musical automata. *Contemporary Music Review*, 18(3):99–112, 1999.

[10] L. Ochs. Devices and strategies for structured improvisation. In J. Zorn, editor, *Arcana: Musicians on music*. Hips Road and Granary Books, New York, 2000.

[11] C. Palacio-Quintin. The hyper-flute. In *Proceedings of the International Conference New Interfaces for Musical Expression, NIME-03*, pages 206–207, Montréal, 2003.

[12] A. Robineau. *Étude sociologique de la musique actuelle du Québec: le cas des Productions Supermusique et du Festival International de Musique Actuelle de Victoriaville*. Université de Montréal, 2004.