A collaboration workflow from sound-based composition to performance of electroacoustic music using «Pure Data» as a framework.

Kyriakos Tsoukalas
Labyrinth
Volos, Greece, 37400
ktsoukalas@ktsoukalas.net

Abstract

This paper describes a workflow for composers, engineers and performers to collaborate, using Pure Data (PD) as a framework, towards the design of electroacoustic musical instruments intended for live performances of sound-based music. Furthermore, it presents some considerations about live performance and ideas of creating collaboration tools, possibly as PD GUI plugins.

Keywords

collaboration, composition, embodiment, software reuse

1 Introduction

By choosing PD as DSP engine and automation software, users are able to program in a graphical programming environment. A programming project approach such as the «incremental-build model» [1], may result in a more rapid development of a musical instrument, based on *early performance testing, reuse of software and progressive refinement* [1] of modules. The graphical programming environment of PD could be used by different users as a collaboration environment that combines a visual representation of the automation and DSP programming within a patch-file.

When the case is that the composer, the engineer and the performer is the same person and assuming that this person is sufficiently knowledgeable in each domain of expertise, then the case seems to be ideal from a transfer of ideas point of view but enervates the energy that will be spent in each domain of expertise, since it will be a portion of the total energy of one person. In the case of an association of a composer, an engineer and a performer, the labor is divided to three people and the case becomes more complicated from a transfer of ideas point of view, but gain in knowledge diversity.

Composers could design data-flow diagrams for the sound processing as planned by their composition ideas. Then engineers could validate and, in collaboration with composers, clarify the different parts of the diagrams. In order to program a sound synthesis engine as planned by a composer's diagram, engineers could develop new and use any available class of PD or external libraries and abstractions to shorten development time.

The performer's interface could be based on a separate device-focused (sensors-equipped, handheld, tangible, etc) (sub-)system, that would communicate with PD. Such an approach should transfer to new projects the dexterity gained with certain devices and custom systems by the performers to succeed in embodiment of control parameters through effective mapping of dexterous gestural abilities. Furthermore, making available the mapping of control parameters for calibration by the performer himself, should help his practice gain in quality.

PD software is used by many electroacoustic composers and engineers, who share resources and publish their work and tools online. Exploring the field of electroacoustic music becomes more fruitful by succeeding in implementing new compositional ideas. The workflow ideas described in this paper aim at helping projects of collaboration in electroacoustic musical instrument creation, run with consistency to the initial idea.

2 A workflow

In figure 1, a workflow is visualized by phases roles, milestones and goals. The goals of the workflow are three: conceptual integrity, reuse of software and embodiment of control parameters. The circle implies a progressive refinement of milestones. The triangle sides separate the workflow divisions per domain of expertise (roles) and the triangle edges point out the milestones that should be accomplished by a collaboration of two different roles.

The use of PD as a framework for the workflow is based on PD being a real-time synthesis engine, having a graphical programming environment, being supported by a wide community offering resources and its Standard-Improved BSD license.

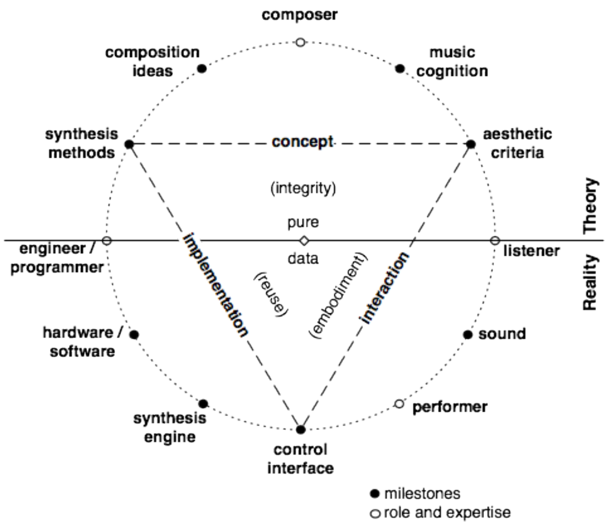

Figure 1 - A workflow

2.1 Composer's role

A composer as the initiator of a project under his compositional concept, is held responsible for the qualitative transfer of his synthesis methods to the engineer, who will try to implement the synthesis methods with a feasible synthesis engine. He is also responsible to discuss matters of aesthetics with the performer.

As proposed by [2], a *semiotic network* between the composer and each listener will be established upon listening or watching a performance. The proposed model of *intention-reception* takes into consideration: a) *The composer's Syntax and Processing of Materials and* b) *The listener's Awareness of situational context and Phenomena of Performance.* During a performance, the composer's syntax will be implemented by an engineer and the processing of materials will be controlled by the performer, while listeners will be stimulated by the context of the performance's phenomena, partially served by the engineer's implementations. Hence, the task to protect the conceptual integrity of the project, emerges.

2.1.1 Aesthetic criteria

A composer moulds his aesthetic criteria as a listener and should be able to analyze listening experiences in order to guide his quest for providing communication signs to listeners.

In a staged performance, aesthetic criteria are not limited to sound alone. There is also a great deal of visual stimuli from performers controlling musical instruments and in electroacoustic music they tend to be partially invisible. The composer and the performer should be able to resolve the variable approaches to aesthetic criteria of the compositional concept, so as to form abstract rules to help with conceptual integrity. Especially, when an ensemble of performers would try splitting compositional tasks.

2.1.2 Music cognition

Analyzing the listener's interpretation of a sound flow as music is a research field of the music cognition discipline. A composer's intentions are carried on a sound flow to face interpretation and by increasing the listeners' perception of his intentions, a composer might gain audience. Therefore, knowledge of cognitive music theory should help composers with knowing their audience.

2.1.3 Composition ideas

Composition ideas refer to the organization of sound in order to produce a sound flow that might guide listeners to make connections between what they listen/perceive and their aesthetic criteria. These ideas regulate how much of the sound synthesis method is planned to fulfill certain aesthetic criteria and how much is planned for experimenting with sound synthesis techniques and eventually form new aesthetic criteria.

2.1.4 Synthesis methods

As the collaboration milestone for the composer and the engineer, any synthesis methods to be used should be outlined by the composer's composition ideas but also evaluated by the engineer who would implement them. The more the reuse of software, the more the need for a collaborative evaluation of this software.

2.2 Engineer/programmer's role

The engineer/programmer is responsible for the quality of the synthesis engine sound quality but also for a qualitative evaluation of the ability of the engine to fulfill the concept's intentions. The role involves straining to combine compositional theory with performance reality, while having to persuade both the composer and the performer for any evaluation findings. He should lead problem-solving.

2.2.1 Hardware and software

Modularity of the system is a key factor for understanding and refining it. The system could take advantage of devices and software as parts of the system. A communication way between PD and any popular device is usually implemented by the PD community.

2.2.2 Synthesis engine

PD is an open-source cross-platform real-time synthesis engine ready to be patched and extended with some new classes.

2.3 Performer's role

The performer is a person that has to deal in real-time with musicology concepts, performance act and controlling gestures. Acquiring dexterity with the control interface is the key to succeed with different tasks in order to fulfill the aesthetic criteria of a compositional concept.

The performer's ability to effect changes to the sound synthesis engine parameters, is regulated by performance behavior (3.1) within the effectiveness of a control interface. Therefore, it is essential for him to be able to embody the different control parameters of a synthesis engine.

2.3.1 Control interface

The development of a control interface might be accelerated by incorporating to a new control system, devices/ interfaces that are already mature and sturdy. The performer may already be dexterous at using any incorporated device/interface. Of course planning for changes is crucial, since it is a goal to progressively refine or even redesign parts of the total system.

Any extra complexity of the control interface is going to cost in performance behavior efficiency and if a performance's context is mostly improvisational, it is more possible that performers will prefer to create musical context, based on what they think they could better accomplish using the current control interface.

2.3.2 Sound

Sound is the product of a synthesis engine, yet it is the principle unit of sound-based music.

2.4 Listener

A listener receives music performance phenomena but he perceives them in his own unique way. Inexperienced listeners may be driven with basic musical context and the way a listener focuses to specific parts of musical context regulates the way he conceives of organized sound as music.

In a staged electroacoustic music performance, even though the practice of reductive listening is promoted by the means of electroacoustic sound production, it is inevitable that there are expectations, especially among inexperienced spectators, regarding the visual stimuli transmitted from the stage (show). Knowing their audience/spectators, should help performers with designing ways to either feed or block such expectations. There is a legacy from acoustic musical instruments, to provide visible evidence of sound synthesis parameter control. But such evidence is of value mostly as being evidence for the live character of a performance.

3 Performance of electroacoustic music

In [3] it is reported that: *«by far the most responses as to why people choose to compose sound-based music were related to the notion of emancipation (freedom).» - i*n two general levels: 1) *Compositional* and, 2) *Expressive*.

Rules of sound organization are formed by analyzing some prototype(s), so as composers could apply them to compositions in order to give the music some popular / widely recognizable characteristics. In a same approach, a performer need to create rules to effectively manage, in realtime, the wide possibilities offered by sound-based music composition on musical context.

While a composer relishes freedom from formalism, aesthetics and freedom of content organisation, a performer is bound to the limits of his ability to control instruments and practice is what would expand those limits.

When possible to split a composition concept in different compositional tasks it could be the case to assign those tasks to different performers of an ensemble. By doing so, an ensemble of less dexterous performers with less fine-tuned musical instruments, might succeed in the implementation of complex compositional concepts and also become aware of musicology concepts.

3.1 Performance behavior

Performance behavior linked with performance context are categorized by [4] in three domains (behavior/context): *1) model/symbol, 2) rule/sign and 3) skill/signal.*

Model-based behavior creates context by switching between usage of rule-based or skill-based behavior. Switching is based on the performer's reasoning about currently perceived sound, awareness of his control actions and consideration of compositional concepts. *Rule-based* behavior creates context by execution of predefined sound synthesis procedures, matched to signs of communication. *Signal-based* behavior requires the performer's continuous parameter control over a signal.

In a “live” electroacoustic music performance there can be a great deal of automation introduced to a sound synthesis engine. Automation, may be used to minimize the parameter control of a sound synthesis engine only to those needed (planned) for effecting changes to the synthesized sound, or in opposition, automation may be used to precompose model-based behaviors.

The synthesis engine's emergence as musical instrument, occurs when the system combines implementation effectiveness and embodiment of parameter control. The performer's behavior regulates the *interruption tolerance* [4] of parameter control, affecting the quality of performance. The rate at which a performer can effect change to the musical instrument is a critical factor in organizing sound at will.

4 PD graphical programming environment as asynchronous collaboration platform

It has been proposed that the graphical programming environment of PD could also be used as the collaboration platform for people working on the same project. Programming in PD using abstractions, sub-patches into sub-patches and scattered code might end-up to a code hardly amenable to change, hence slow down refinement of modules. Modules of better cohesion might be produced, if programming is supported by some simple collaboration tools that would be integrated into PD's graphical programming environment. PD version 0.43 offers the capability to implement tools as GUI plugins, while dynamic patching and PD's data structures could help with object creation and graphs.

Those tools could make some patch changes more transparent and aid collaborators in keeping track of valuable information. Generation of some useful info or diagrams could facilitate programming and communication. Some specific thoughts about possible functions of collaboration tools:

1. “Post-it”: Query-able text editor with history and possibly image/video support.
2. “ToDo”: Reminder info and warning popup per (sub-)patch.
3. “Structure chart”: Diagram of sub-patches and abstractions included in a patch and its abstractions with tracks of any “un-patched” connections (like send-receive).
4. “Control log”: Data-mining viewer/log of input data and user-selected variables (for example to facilitate investigation and/or experimentation with control parameter input-data patterns, performer's reaction time or data-mining efficiency).

5 Conclusion

Using PD is not only a way to lower costs but also to learn and keep up with newer and even state of the art audio processing.

The workflow proposed, may be of value mostly to beginners in sound-design and composition of electroacoustic music. It plans for a division of labor within roles and promotes team collaboration. The adopted *incremental-build model* [1], allows for ephemeral harvesting, ensures a functional musical instrument for yet another performance and provide some proof against budget/scheduling changes or problems.

PD's graphical programming environment could be enhanced with asynchronous collaboration tools, possibly as GUI-plugins, to aid collaboration within PD's graphical programming environment.

Future work is to investigate a small number of teams following the principles of the proposed workflow and to implement collaboration tools after a relevant research and more careful consideration.

6 Acknowledgements

My thanks to the online PD community members for sharing work and knowledge.

References

[1] Frederick P. Brooks, JR.: *“The Mythical Man Month”,* pp. 267-273, Adisson-Wesley 1995.

[2] Adam Basanta: *“Potentialities of Meaning”,* Simon Fraser University essays 2008.

[3] Rob Weale: *“Exploring our relationships with sound-based music”,* Music studies Network International conference 2008.

[4] Joseph Malloch, David Birnbaum, Elliot Sinyor, Marcelo M. Wanderley: *“Towards a new conceptual framework for digital musical instruments”,* Proceedings of the 9th int. conference on digital audio effects, Montreal, Canada 2006.