San Jose State University
SJSU ScholarWorks

Faculty Publications | School of Music and Dance

January 2003

Compositional and Programming Issues Within Lyra, a Fully Interactive Performance Environment For Violin and Kyma System

Brian Belet
San Jose State University, brian.belet@sjsu.edu

Follow this and additional works at: https://scholarworks.sjsu.edu/music_dance_pub

 Part of the Music Commons

Recommended Citation

Brian Belet. "Compositional and Programming Issues Within Lyra, a Fully Interactive Performance Environment For Violin and Kyma System" *Proceedings of the 2003 International Computer Music Conference* (2003).

This Article is brought to you for free and open access by the School of Music and Dance at SJSU ScholarWorks. It has been accepted for inclusion in Faculty Publications by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Compositional and Programming Issues Within *Lyra*, a Fully Interactive Performance Environment For Violin and Kyma System

Brian Belet

Director, Center for Research in Electro-Acoustic Music
School of Music and Dance, San Jose State University
email: bbelet@email.sjsu.edu

Abstract

Meaningful real-time interaction between human performers and computer processing is an important aesthetic issue for many composers. With the advent of computer systems that are actually fast enough to permit real-time algorithmic sound realizations based on the analysis of live performance data the issue now facing composers is how to utilize these tools in a meaningful and artistic way.

Lyra, composed in 2002 for violin and Kyma, is an environment in which the acoustically generated sounds and the computer generated and processed sounds are mutually dependent upon each other for a unified ensemble performance. All of the computer music layers are generated in real time using the violin music as direct audio and as analyzed input data for processing, resynthesis, and other parameter control. The violin music is also affected by the computer music output through performance instructions that invite response and improvisation. The violinist is also able to control macro time and gestural synchronization with the computer sound layers. The aesthetic goal is to create a performance environment that permits maximum flexibility for the human performer with a great deal of linear independence for both violin and computer, while still maintaining a very high degree of unity and ensemble interaction.

1 Introduction

Meaningful real-time performance interaction between live performer and computer sound generation is truly possible with Kyma 5. The composition *Lyra* utilizes the real-time software and hardware processing power of Kyma in a fully interactive way. All of the computer music layers are generated in real-time performance and are derived from the live violin audio data, which are processed directly as audio signals and also analyzed to derive independent frequency and amplitude data (FFT) for resynthesis and other parameter control. As a result the computer music is completely dependent upon the violin music during performance.

The violinist is invited to deviate from the fully composed score in order to react to the computer sounds, to improvise upon any notated gesture, and to perform notated gestures in a variety of performance orders (including repetition and omission). The composer's intention is to create a performance environment in which the violin and computer layers are mutually dependent upon each other in a true ensemble context, while still preserving maximum independence and performance freedom within each respective layer. These mutually dependent aspects of the composition will be described below and demonstrated using the Kyma system.

2 Kyma.5 digital synthesis and processing system

Composers have been interested in utilizing real-time computing in live performance for many years: not just triggering stored samples and hardware processors during performance, but actual sound generation that relates musically and aesthetically to acoustic instruments and voices. The combination of Kyma's very efficient software code and the Capybara's extremely fast parallel DSP processing enables a composer to create performance environments in which human and computer performers can fully interact and generate music that is mutually dependent and responsive in a genuine performance ensemble context. *Lyra* was composed to explore these capabilities.

While the Kyma digital synthesis system, with its linked software and DSP hardware, has been in use for over ten years, several significant changes were

introduced with Version 5 of the Kyma software (released in July 2001). For this current project new features utilized include the TimeLine, used for structuring macro time and as a visual performance interface, and the WaitUntil Sound, used for performer control of the computer clock time.

3 Compositional issues for *Lyra*

The traditional paradigm of live performer with taped (or otherwise fixed) electronic sounds places the human performer at a practical disadvantage. If anything goes awry during performance it is the human performer who is left hanging unsupported and in the public perception of error. Whether the performer misses an entrance with the fixed sounds or the audio playback suffers a mechanical breakdown, it is the performer on stage who is placed in the unfair position of being out of sync with the unforgiving prerecorded electronic layers. *Lyra* was designed to place the human performer back into a more realistic and friendly position of being an organic part of the overall performance ensemble. The violinist is able to control aspects of macro computer clock time within Kyma, using the WaitUntil Sound, which can use either an amplitude threshold or a specific frequency test to set either positive or negative signal controls. For *Lyra* the violin's initial attack creates a positive amplitude control value that starts the TimeLine clock for immediate computer sound generation and synchronization (Figure 1). The flexible performance path through the notated score also gives more performance control back to the human performer.

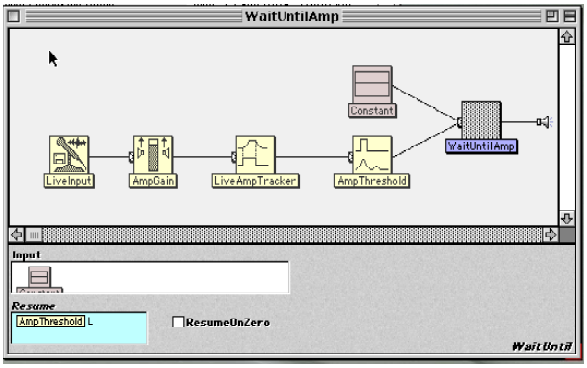

Figure 1. Violin amplitude is tracked, and a positive control signal is generated when the tracked amplitude exceeds the threshold value.

The composition is unified by having all of the computer music relate to the violin music, both directly and indirectly. There are no pre-recorded or pre-synthesized sounds stored in memory as physical samples or deterministic algorithms. Without the violin playing there is no computer music. The initial attack of the performance is controlled by the violinist: even though the TimeLine is compiled and running, no sound occurs until the violinist begins the performance. The violin sound serves as audio input into the Capybara DSP hardware, and the computer sounds begin immediately in synchronization with the violin. The violinist controls the overall texture of the computer sounds throughout the performance: density, gestural speed, and energy "feel" remain consistent between the two performance sources.

4 Programming issues for *Lyra*

The audio signal from the violin is routed into the Capybara DSP hardware as a direct audio input. This audio signal is processed directly through several algorithms, including granulation clouds, multiple random delays, and ring modulation with delayed retrograde output. Figure 2 shows the processing structure of a Sound that utilizes the violin amplitude data as a processing parameter within a Vocoder synthesis algorithm (top of next page).

The violin signal is also analyzed for its separate frequency and amplitude data, which are used for direct resynthesis and also as indirect control for other processing algorithms (Figure 3, on next page).

The end result is that all of the computer music is generated in real time performance using the live violin data as input parameters for Kyma Sounds.

5 Demonstration of algorithms used for *Lyra*

The algorithms cited in the figures above will be demonstrated, along with additional algorithms used in the composition *Lyra*. These include nonlinear wave shaping with delay, a five-delay structure whose mixed outputs create dynamic cascading interference gestures, a multiple random delay, ring modulation utilizing retrograde delay, and a granulation cloud in which both cloud density and grain duration parameters are controlled by the violin amplitude.

The various computer processing algorithms are organized into a performance environment using the TimeLine to control macro time. This visual interface is quite efficient for performance monitoring by both the performer and the audio mixing person (Figure 4, on next page).

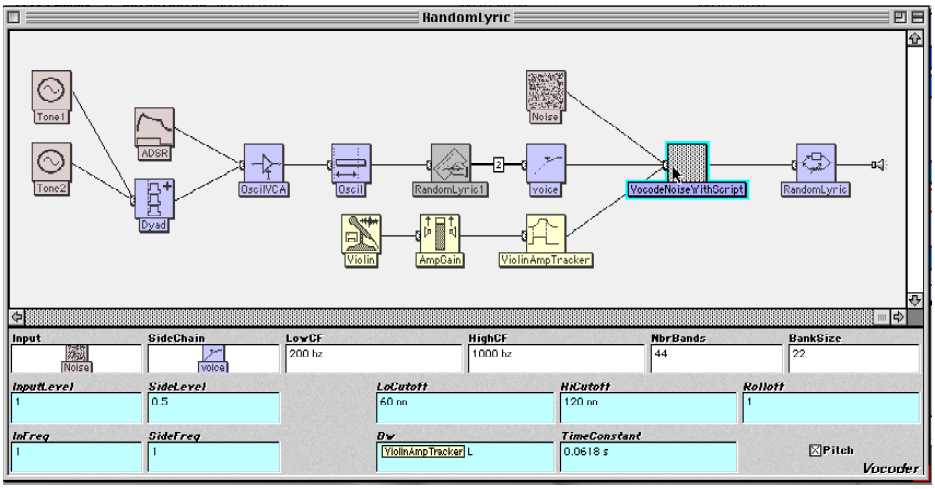

Figure 2. A background lyric layer is stochastically generated using the composer's program COMP2 (located within the RandomLyric Script Sound). The violin amplitude is tracked and that value is used to control the bandwidth of the Vocoder superSound.

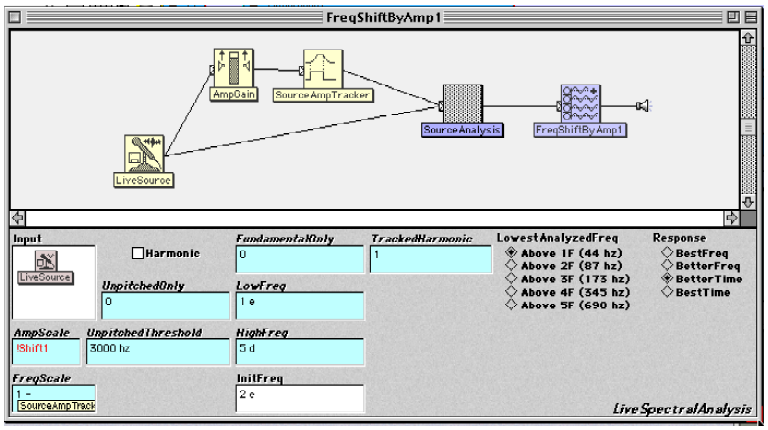

Figure 3. The violin audio signal is analyzed within a LiveSpectralAnalysis Sound. The violin amplitude is separately tracked and used as an inverse function for the analysis frequency scale. These data are used to resynthesize a new violin layer, whose frequency is inversely modulated by the violin music's amplitude.

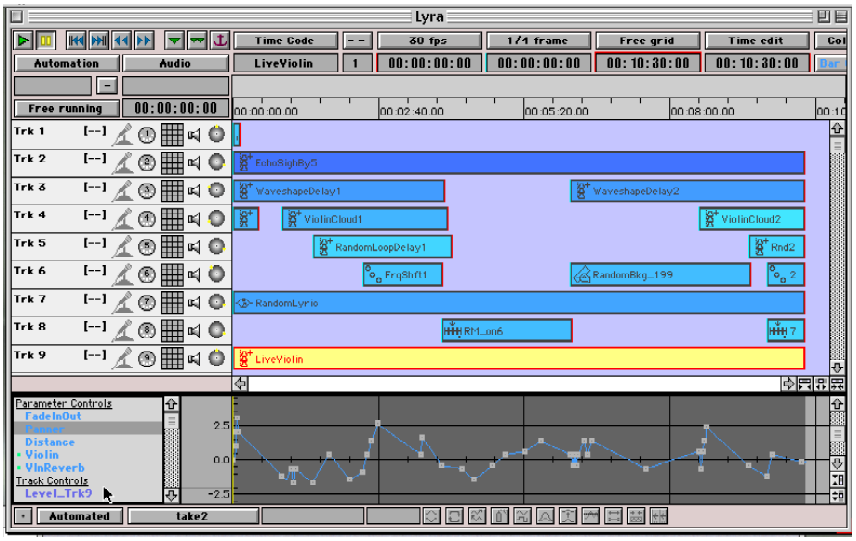

Figure 4. TimeLine for *Lyra*.

The WaitUntil Sound is located on Track 1 for isolated and easy visual identification. In this usage the Sound is assigned an amplitude threshold value, which is triggered by the violin's initial attack gesture to start the computer processing clock time. For most performances the computer requires no intervention by the audio person. Various real-time parameters have been added on screen for use during the sound check (not shown here): this allows various levels and densities to be adjusted without recompiling the code. To date none of these parameters have had to be adjusted during an actual performance. The DSP status window is also present to permit monitoring of the DSP processor usage during performance. While this is more important as a debugging tool during development of the composition, this composer prefers to have that data present during performance.

6 Conclusions

Lyra was composed in 2002 to fully integrate the human performer and computer processing as described above. It was composed as an experiment to examine if this level of mutual interaction was actually possible. Through a combination of efficient and composer-friendly software code and powerful DSP hardware this experiment has proven successful. Since completing *Lyra* the composer has completed a new composition for piano and Kyma (2003). *(Disturbed) Radiance* extends this experiment further, exploring more ways to integrate the human performer and the computer processing in performance.

7 Acknowledgements

This research and composition is supported by the School of Music and Dance, and by the College of Humanities and the Arts at San Jose State University. Countless hours of technical support and artistic encouragement have been provided by Carla Scaletti and Kurt Hebel of Symbolic Sound Corporation.

References

Belet, B., 1991. "Proportional Recursive Stochastic Composition Using *COMP2*, a Smalltalk-80 Composition Program Within the Kyma Digital Synthesis System." *Proceedings of the 1991 International Computer Music Conference.* International Computer Music Association, pp. 513–16.

Belet, B., 1992. "Toward a Unification of Algorithmic Composition, Real-time Software Synthesis, and Live Performance Interaction." *Proceedings of the 1992 International Computer Music Conference*. International Computer Music Association, pp. 158-161.

Garnett, G., 2001. "The Aesthetics of Interactive Computer Music." *Computer Music Journal* 25(1): 21-33.

Scaletti, C., 2002. "Computer Music Languages, Kyma, and the Future." *Computer Music Journal* 26 (4): 69-82.

Scaletti, C., 2000. *Kyma.5 Walkthrough: A Tutorial Introduction to Kyma.5*. Symbolic Sound Corporation.

Scaletti, C., 1989. "Composing Sound Objects in Kyma." *Perspectives of New Music* 27(1): 42-69.

Stockhausen, K., 1996. "Electroacoustic Performance Practice." *Perspectives of New Music* 34(1): 74-105.

Supper, M., 2001. "A Few Remarks on Algorithmic Composition." *Computer Music Journal* 25(1): 48-53.

Vaggione, H., 2001. "Some Ontological Remarks about Music Composition Process." *Computer Music Journal* 25(1): 54-61.