

Real-Time Generation of Music Notation via Audience Interaction Using Python and GNU Lilypond

Kevin C. Baird
University at Buffalo
222 Baird Hall
Amherst, NY 14221
01-716-564-1172

kcbaird@world.oberlin.edu

ABSTRACT

No Clergy is an interactive music performance/installation in which the audience is able to shape the ongoing music. In it, members of a small acoustic ensemble read music notation from computer screens. As each page refreshes, the notation is altered and shaped by both stochastic transformations of earlier music with the same performance and audience feedback, collected via standard CGI forms.

Keywords

notation, stochastic, interactive, audience, Python, Lilypond

1. INTRODUCTION

No Clergy currently runs on a Debian GNU/Linux system. The ideal performance setting has one person serving as the “conductor” logged in to a shell on the server. It requires a small ensemble of acoustic performers playing any sort of monophonic pitched instrument. Each performer needs a web browser on which to view notation. There should also be one or more stations with web browsers pointed to the CGI forms used by the audience to react to what they are hearing.

Karlheinz Stockhausen's *Klavierstücke* and the work of the late Earle Brown, especially *Available Forms I & II*, were especially influential on *No Clergy*. [4,11] From them, it uses the idea of a random access ordering of material played by acoustic performers. *Klavierstücke* gives such power to the performer. Brown's work gives such power to the conductor. *No Clergy* gives it to the audience.

2. INITIAL PAGE OF NOTATION

2.1 Generation of Notation Markup

To run the piece, the “conductor” executes a bash wrapper, whose technical details are discussed elsewhere. [1,2] This initial page of notation is intended to be fairly “neutral”, using a flat distribution of pitches, durations, articulations and other musically-significant data across either each instrument's range or some generic global ranges. The program also reads a configuration file for variation of these initial parameters, allowing different starting points for performances.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Nime'05, May 26-28, 2005, Vancouver, BC, Canada.

Copyright remains with the author(s).

2.2 Storage as MusicXML

In order to perform stochastic transformations on the notation, *No Clergy* needs some method of storing musically significant information about each page of notation. I chose Recordare's MusicXML, described by them as “a universal translator for common Western musical notation from the 17th century onwards. It is designed as an interchange format for notation, analysis, retrieval, and performance applications.” [10] Each most recent XML file is accessed during the generation of subsequent pages of notation, while older files are compressed and stored.

2.3 Rendering of Notation Markup

After the “conductor” has executed the bash wrapper script, it will then have created one GNU Lilypond markup file for each instrument. Lilypond is a Scheme-based music typesetting program which uses a TeX-like backslash notation for formatting commands. It is inspired by “the best traditional hand engraving”, and outputs to several high-resolution graphics formats. [7]



Figure 1. Initial page of notation for trumpet

Figure 1 shows an initial page of notation generated for a trumpet, as output to PNG for viewing in a web browser. The performers then play their individual pages of notation. They are not required to stay in sync with each other.

3. AUDIENCE FEEDBACK

As the performers play, audience members are encouraged to respond as they see fit, using a CGI form specific to a given instrument. Audience members are able to values corresponding to points along a continuum (shorter vs. longer durations, louder

vs. softer volumes, etc.) and values representing allowable variation from these points. This second type of values can be thought of similarly to the Q of a filter, or a metaphorical “strength of conviction” of the first type of value.

Audience members are therefore able to affect both the overall direction of change (shifting articulations toward *staccato*, for example), as well as the range of variation. Narrow variation causes extremely soft dynamics to cluster around the minimum value of *ppp*. Wide variation from *ppp* allows more notes with *mp* or even *f* dynamic indications, even as the overall dynamic center point remains extremely low.

4.SUBSEQUENT PAGES OF NOTATION

For the 2nd and later sets of pages for each instrument, the conductor executes a different bash wrapper script, which generates Lilypond markup via Markovian transformations of the preceding pages, rather than using a uniform distribution.[5] This markup is also shaped according to the audience feedback. An example of such output after several successive runs of the piece is shown in Figure 3. As you can see, the feedback has resulted in a thinner texture, with a narrow dynamic range around *ff*.



Figure 2. Transformed page of notation for trumpet

5.AESTHETIC MOTIVATIONS

No Clergy is an attempt to explore the boundaries between composer and audience. By providing tools for the audience to participate in shaping the outcome of musical performances as they occur, I hope to demystify the “priesthood of the composer”. Performance scenarios in which audiences feel comfortable participating in events are therefore critical for the success of the piece.

Audience familiarity was a key reason for choosing CGI forms as the main interface for the audience: web browsers are very familiar to many non-musicians, and they are often used to inputting data in forms. To the extent that this encourages the audience to participate, it adds to the potential success of each performance.

6.OTHER PERFORMANCE OPTIONS

Currently, the program is run manually by the conductor. Since it consists of command-line scripts in a Unix-like environment, setting up scheduled executions with periodic resets back to initial page conditions is trivial.

The piece as it exists is also intended for performance by real acoustic musicians. Were *No Clergy* to be demonstrated as more of an installation, making such demands of performers becomes more onerous. Other output options such as MIDI or running through Csound, Max/MSP, or PD may be added in the future. Especially for a musically literate installation audience, visual presentation alone is also a feasible option.

7.ACKNOWLEDGMENTS

Thanks to Cort Lippe, Director of the LeJaren Hiller Computer Music Studios, and Jeff Higginbotham, Director of the Communication and Assistive Devices Laboratory, both at the University at Buffalo. Thanks also to J.T. Rinker, Ben Harris, and Susan Fancher for performer feedback.

8.REFERENCES

- [1] Baird, Kevin C. “No Clergy: Real-Time Generation and Modification of Music Notation.” In *Proceedings of the 2005 Spark Festival* (University of Minnesota, Minneapolis, Minnesota, Feb 16-20, 2005), 40-41.
- [2] Baird, Kevin C. “Generating Music Notation in Real Time.” In *Linux Journal*, 128 (Dec. 2004), 72-76.
- [3] Baird, Kevin C. “Multi-Conductor: An Onscreen Polymetrical Conducting and Notation Display System.” In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)* (McGill University, Montreal, Canada, May 22-24, 2003). McGill University Faculty of Music, Montreal, Canada, 2003, 211-212.
- [4] Brown, Earle. *Available Forms I & II*. Associated Music Publishers, Inc., 1962.
- [5] Dodge, Charles and Thomas A. Jerse. *Computer Music*. Schirmer Books, 1985.
- [6] Hiller, Lejaren A., Jr. and Leonard M. Isaacson. *ILIAC Suite for String Quartet*. New Music Edition, 1957.
- [7] Nienhuys, Han-Wen and Jan Nieuwenhuizen. *GNU Lilypond*. <http://www.lilypond.org>.
- [8] Recordare, Inc. *MusicXML*. <http://www.recordare.com/xml.html>.
- [9] Stockhausen, Karlheinz. *Klavierstücke*. Universal Edition, 1954.