CALDER’S VIOLIN: REAL-TIME NOTATION AND PERFORMANCE THROUGH MUSICALLY EXPRESSIVE ALGORITHMS

Richard Hoadley
Digital Performance Laboratory
Anglia Ruskin University
Cambridge UK

ABSTRACT

Notation is a central issue in modern western music. Composers have often sought ways of expanding and refining the functionality of notation and, in doing so, have re-shaped the music that they were originally aiming to describe. Other musical traditions have used notation very differently; some have used no notation at all, and have thus created contrasting musical experiences.

The role that electronics and computers have played in music has also influenced the nature and function of notation. More traditional ‘live’ notation of note/pitch-based music generated algorithmically has proved particularly problematic: musical notation is itself a very complex subject. Composers and technologists have instead used libraries of images, algorithms for the pre-generation of material or simplified notations that can be used as the basis of more improvisatory performances.

This paper presents work involving the live presentation of ‘traditionally precise’ music notation created through algorithmically generated material. This notation can then be performed by a human musician alongside computer-generated diffused sound or other ‘real’ musicians. Technologies used include the SuperCollider audio programming environment and the INScore notation project with the Open Sound Control protocol used to communicate between them. As well as providing a fascinating musical experience, the process highlights a number of issues concerning performance practice, instrumental technique, rehearsal, time and timing, as well as the nature of notation itself and its relationship to improvisation.

1. INTRODUCTION

One of the more divisive issues in electronic music today involves its relationship to live performance: the move from ‘object to dynamic system’, discussed by Chadabe [3], Collins [4, 5] and Ariza [2], perhaps reflecting the way in which music notation is a mediating element allowing dynamic interchange between composer and performer. Hudak et al. [12], attempt to understand notation from the perspective of a functional programmer, indentifying many of the ‘limitations’ of ‘common practice notation’: finding it frequently ‘deficient, inconsistent and redundant’ and pointing out perennial issues such as the number and type of tuplet that should fit into a given duration. More obscurely they claim that ‘traditional notation is unable to adequately capture a composer’s intentions’, and that

‘traditional music is biased towards music that is humanly performable’ (their italics). This is however, unsurprisingly, ‘an obstacle when trying to notate music intended for computer performance’. A musical view here might hold that it is not the logicality or otherwise of the relevant notation system that is important, but its familiarity and the skill with which the composer is able to utilise it for his or her own expressive purpose.

Algorithmic music has been one of the means by which composers of instrumental music have been able to express themselves through mechanical and computer processes. Why they should wish to do so is beyond the scope of this paper, but some speculations have been proposed by Rowe [15] and Loy [13]. In the author’s case, foremost in these motivations is the role algorithms can play in the understanding of the musical process, as well as the fact that they allow us to ‘play’ with musical ideas and patterns.

In comparison to long-standing developments, both in algorithmic software for computer-aided composition (CAC) and ‘engraving’-based music notation software, development in software allowing real-time notation has been less blessed - particularly frustrating as it allows the use of truly live and interactive musical patterning in the electroacoustic context.

2. RELATED WORK

There is significant literature concerning algorithmic or generative music. Nick Collins discusses not only the types of music these terms describe, but provides a distinction between the two [4]. Much of Collins’ work in this area is concentrated on the relationship between large-scale musical form and the individual algorithmic function itself: what he has termed the bottom-up approach. An alternative ‘top-down’ perspective is investigated in Hedelin [10], but while working on Calder’s Violin I avoided these issues, concerning myself “only with the generation of basic musical materials rather than self-contained pieces”, retaining “the right to deal with larger concerns in the usual intuitive manner” [5] – a very definitely ‘top-down’ approach.

Developments in CAC software have also generated a significant literature: the increasing popularity of middle and higher level tools such as MaxMSP, OpenMusic and SuperCollider has meant that the generative composition process itself has become increasingly accessible to less technically-minded composers.
Real-time notators are rare in comparison with more ‘engraving’ oriented software (Lilypond, Sibelius, Finale, Noteability, etc.), but for a number of years there have been attempts at filling this gap. These include Wulfson et al, [16], Freeman [8], McClelland and Alcorn [14], Didkovsky & Hajdu [6], Agostini and Ghisi [1]. These systems include methods for defining and ‘projecting’ notation live. All, except for MaxScore and the Bach Project, deliberately use a limited sub-set of standard notation: the latter two use live notation as a part of more general CAC systems rather than as dedicated live notators.

One reason for the above mentioned gap concerns the complex nature of music notation itself. This is difficult to deal with under the best of circumstances and problems can arise particularly in the context of live performance. An example of this is the use of glyphs or other markings requiring a knowledge of the length of the phrase in question in order to correctly calculate length, width or configuration; this knowledge doesn’t necessarily exist until the phrase or function has completed its evaluation.

3. IMPLEMENTATION

Some parts of the music for Calder’s Violin are based on interactive music written for the music-dance production of Triggered performed in London in June 2011 [9]. This music was written (in code) as an attempt at creating liveness and controllable unpredictability when combining scheduled algorithms with the interpretation of data acquired from dancers’ movements, gestures and touch. As a key musical sound used was that of a synthesized piano sounds, the possibility of a conversation between ‘live’ piano, and algorithmically generated piano emerged. Software for generating live notation include MaxScore [6], Bach [1] (each for MaxMSP) and INScore [7]. INScore, while still under development, allows OSC control and is a software environment optimized for external control.

4. METHOD

In Calder’s Violin an algorithm is usually a usually high-level function, and there are about 270 of these functions scheduled in the composition. The scheduling involves precisely timed events sometimes coloured with varying levels of unpredictability (evaluate the function in 2.0 plus the random of 1.0 second, for instance). There are also a significant number of notational and ‘silent’ functions – the latter, for instance, making sure some previously evaluated functions have been terminated. The piece otherwise follows a broadly ABA’ structure where A is delicate, florid and decorated; B is faster and more rhythmic.

5. NOTATING ALGORITHMS

An example of an algorithm from the piece is the function called ‘~chord06’:

```
~chord06.value(62+(6.rand), [ 0, 3+(4.rand), 4+(4.rand), 3+(4.rand) ], 0.0, 0.002, 48, 1.2, 0.0, true, true, 1, "chord");
```

Run six times, this produces a series of six four-note chords, to be played and/or displayed according to various other parameters. At the time of writing they generate the set of chords in Figure 1 and subsequently the a line of notes in Figure 2.

![Figure 1](image1.png)

**Figure 1.** A sample output of the function ~chord06 in INScore

![Figure 2](image2.png)

**Figure 2.** A single line output of the function ~chord06 in INScore

The SuperCollider language (sclang) is then used to trigger the audio server (scsynth) into ‘playing’ the audible part of the synthesised composition. At the same time, the sclang data is converted into the relevant OSC messages that control the INScore Viewer application. This provides a fast, optimised system for the notation not just of ‘standard’ western notation, but a number of other graphic and textual elements. The below message is the final, formatted OSC message sent as a result of processing a part of the sclang code included above, which is then interpreted and displayed in the INScore viewer (Figure 3):

```
netAddr.sendMsg("ITL/scene/myScore1/", "set", "gmn", [{ f1/4, g#1/4, d2/4, g#2/4 }]);
```

![Figure 3](image3.png)

**Figure 3.** A typical output as displayed to the performer
6. ISSUES ARISING FROM THE COMPOSITION AND PERFORMANCE

6.1. Rehearsal and preparation

Most performers require rehearsal in order to engage with the music as well as to develop their interpretations of the piece at both macro and microscopic levels. Practice may begin with a review of any background material concerning the piece or composer, before attempting to develop a general feeling for the overall shape - how long is it? Which style does it use? - finally concentrating on the work of getting the notes right.

This process is not available in its entirety to the performer in Calder’s Violin. In this particular piece, the macroscopic structure remains similar in each performance; in that respect it is not materially different from the performance of a composition with fixed notation. However, the detail is significantly different each time.

Although no pianist, I have performed other pieces based on similar principles and can sympathise with the performer’s situation. However, there are advantages in being given a ‘finite’ part rather than just being told to ‘improvise’ in a particular way, with maybe a scale provided as an example. Classical performers are used to providing, quickly and efficiently a confident, fluid performance, and it is often this confidence that can carry a performance. To help with this I have tried to ensure that the music the software generates is either not too difficult technically, or is music that can be ‘improvised’ with some ease (swirling chromatic passages, for example).

6.2. How much to display and when to display it?

While I had taken some time in preparing for my first rehearsals with Mifune, it quickly became clear that many of the presumptions I had made about the display of the generated notation were inaccurate. A fixed, physical score contains much information that is not immediately obvious: an instant overview of the number, size and content of the pages.

If you send INScore a ‘note’ message it will default to displaying this in the absolute centre of the page. If you then send a second message with the first note and a new one, this two-note phrase will be re-centred, and so on. One of the choices, then, is whether to send a phrase so that it appears one note at a time, gradually expanding from the centre of the page, or whether to send the material one phrase, or even one ‘page’ at a time. In the event there are cases of each usage, the choice of which to use depending very much on the nature of the music and the performance requirements at that point in the piece.

6.3. How soon is now?

An as yet unresolved issue is how much ‘notice’ of an upcoming event should be given to the performer. This isn’t a matter of rehearsal as discussed above, but literally of how much time is required for someone to notice, take in and react to the appearance of a musical phrase. If you look at the video made of the main rehearsal you can see Mifune’s nerves and concentration as she readies herself for whatever the programme will send her [11].

In general it seems that a delay in the order of one and a half seconds is sufficient, although sometimes more would be an advantage, particularly when the generated music is slightly more complex. For the software to allow arbitrary lengths of delay would require a modification of the scheduling functionality. At present, algorithmic data is written to ‘live’ arrays which maintain their content for the duration of the algorithm. Subsequently, if another set of data is generated, that previous set is over-written and is no longer accessible. If a scheduling delay is then introduced that is longer than the timed gap between evaluated instances of the function in question, the rescheduled function will wrongly use the more recently generated data. In order to allow arbitrarily long delays, all functions would have to write to specifically stored data addresses and these, then, would be used for ‘playback’. While there is nothing wrong with this in essence, it does call into question the nature of the idea of ‘liveness’. In reality a performance using stored data, even if the data is algorithmic, is no different from any other type of fixed media playback. Would the issue be different if data addresses were overwritten immediately they are played back, so retaining their anonymity? But in reality, does over-writing or deleting make any real difference to the situation? The problem is a difference in the way in which wet-ware works in comparison to hard and software, in the particular case when also combined with the challenged of an un-fixed score.

Does it matter when the data is generated?

6.4. Other issues and questions

Other questions involve whether to include visualisations of the accompanying synthesised piano material in order to help with coordination and prediction. With the addition of more instrumental parts this becomes more important, but for this piece it proved unnecessary. In this sense, the violin part of Calder’s Violin remains little different from its papery counterpart.

A ramification of not including visualisations of the computer part is that there cannot be something acting as a “pulse” or “tempo” that might be shared by two people (or a person and a computer). The violin part is notated in accordance with some precise durational algorithms that are designed to display material in ways that are clear and expressive although modulatable. Should the experience of other more professional performers require it, but of course duration is only one factor in the production of tempo. Mifune seemed to respond naturally to these inconsistencies in a way that made me feel confident about the intuitive methods I had used in describing the notation’s inherent vagaries,
particularly the subjective interpretations of relationships between pulse and tempi. The computer music sometimes sounds as if it has a pulse, but this is an illusion that can be ignored if necessary without the ‘normal’ ramifications that might occur in human performance.

A final point regards common annotations, such as dynamic and phrasing marks. While not impossible, these are somewhat cumbersome to implement, while also being quite difficult to see when dynamically displayed on a computer screen. As INScore is able to display custom text in a variety of sizes, I decided to use this feature, if only as a temporary measure. This also enables me to include basic cueing instructions, such as ‘play on next beat’ or ‘play now’.

7. CONCLUSIONS AND FUTURE WORK

One of the most significance issues arising from this work is the matter of timing, as mentioned above in terms of scheduling delays, but also regarding the matter of synchronization, whether between instrumentalist and computer or between more than one instrumentalist. A number of new works involving more than one human performer are being composed which should help to resolve this issue at least in part.

Related to this issue is how much, if any, ‘history’ to include, and how much access should be given to it. In other projects it is accepted that the accumulation of historical data from sessions will be useful if not essential for the purposes of analysis and, if necessary, recreation. In creative performance, though, how ephemeral should this data actually be?

How ‘hidden’ should the processes behind live notation be? Should the notation be projected as a part of the performance? Are these details relevant, interesting or simply distracting to an audience? How would performers feel about having their notation displayed publically in this way?

I am personally interested in integration of physical gesture with live notation: there are roles in creative, educational and therapeutic environments.

Finally, for those who are interested, the author is currently preparing a SuperCollider class for use with the INScore Interactive Augmented Music Score. It is expected to be released in 2012 from: http://rhoadley.net/software/inscore/

8. REFERENCES


