

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 very different uses for notation; some have no use for it at all; each approach creates 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

This paper is three things: a case study of how I tackled the role of real-time notation in a specific piece, an aesthetic analysis of the role of real-time notation in music performance and a description of the system through which this specific composition (and subsequent ones) have come about.

In the context of this paper I use the term 'real-time notation' (or 'live notation') to refer to notation that is generated as the performance progresses and where this process is itself considered to be of central importance in the composition.

Algorithmic composition provides me with an invaluable insight into the creative process; it also enables me to generate novel musical ideas automatically through definable processes. Because the

latter are generated by chance as well as choice, the responsible algorithms can be structured to have features, ideas and patterns that are new to me. It is also advantageous that such generated material can be easily auditioned.

Although such a system may be initially developed in association with a specific composition, it inevitably contains functions and processes that can be useful in future work; they are separate from the immediate process of composition. Work on *Calder's Violin* has also resulted in my own development of a SuperCollider class to facilitate communication between SuperCollider and certain aspects of INScore. These resources have already been used to create new compositions *Fluxus* and *Fluxus Tree* [16]; in the case of the latter including experimental methods for the generation of notation through physical movement.

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. [18], attempt to understand notation from the perspective of a functional programmer, identifying 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 logicity 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 [22] and Loy [19]. As mentioned above, in this 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 [14], 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 to fill this gap. These include Wulfson et al, [24], Freeman [8, 9], McClelland and Alcorn [20], Didkovsky & Hajdu [6], Hajdu [13], Agostini and Ghisi [1]. These systems include methods for defining and ‘projecting’ notation live. All, except for MaxScore/JSML and the Bach Project, strategically use limited sub-sets of standard notation: the latter two use live notation as a part of more general CAC systems rather than as dedicated live notators. My use of INScore exploits the programme’s light-touch and speed, with the result that unlike many systems, notation can be displayed anywhere in any configuration on a full screen, at any size and virtually instantaneously. While being of inherent value, this feature has itself highlighted some interesting possibilities. Cheryl Frances-Hoad, a collaborator in *Fluxus Tree*, said that when presented with too much material presented too quickly she found it interesting to intuitively ‘average’ the notation based on where particular phrases appeared on the screen, a creative exploitation of behaviour that might otherwise be considered anomalous [11].

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 [10]. This music was written (in SuperCollider code) as an attempt at creating liveness and controlled 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, the possibility of a conversation between ‘live’ piano, and algorithmically generated piano emerged. Software for generating live notation includes MaxScore [6], the Bach Project [1] (each for MaxMSP) and INScore [7]. INScore, while still under development is a software environment optimized for external control via OSC. INScore’s abilities in symbolic music notation rendering rely on the GUIDO Engine and the MusicXML library.

4. METHOD

In *Calder’s Violin* an algorithm is usually a high-level function, and there are about 270 evaluations of these functions scheduled in the composition. The scheduling involves precisely timed events sometimes coloured with 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.

I call these algorithms ‘musically expressive’ because the principal motivation in their design is to emulate my own ideas and gestures: imaginings that are traditionally expressive in musical terms. This iterative process of imagining, implementing, re-imagining, re-implementing, etc., itself plays a very important role in the development of both algorithm (function), musical gesture and indeed the musical context in which these gestures are to occur, as it does in more traditional notation-based composition. For me, all of these components work together as musical composition. In order to make the gestures produced by the algorithms fully a part of the composition, elements (arguments) were included in order to increase what I would term their ‘expressivity’: controls on note duration (tempo/rubato), amplitude, note length (articulation), etc. In subsequent projects, these functions might then be extended in a way that reflects both the extension of an algorithm’s functionality in software and the musical development of a melody, a phrase’s shape or the nuance of a harmony.

5. NOTATING ALGORITHMS

An example of a simple 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 line of notes in Figure 2.



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



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

In contrast Figure 3 shows a more expressive melodic output generated by the function ~doMelBit(0.1). The melodic material is a combination of fragments taken from a generic melodic row and more random figures. Note durations are restricted to a proportionate and metric range of possibilities. The single argument controls the number of rests present in a generated phrase: in the composition the function generates the violin part as it dies away to the end:



Figure 3. a single line output of the function ~doMelBit

The SuperCollider language (sclang) is 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 [7]. The below message is the final, formatted OSC message sent as a result of processing a part of the sclang code included in the first example above, which is then interpreted and displayed in the INScore viewer (Figure 4):

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



Figure 4. a typical output as displayed to the performer

6. ISSUES ARISING FROM THE COMPOSITION AND PERFORMANCE

Calder's Violin has to date been publically performed by violinists Mifune Tsuji and Marcus Barcham-Stevens for concerts that were a part of the Cambridge Festival of Ideas in October 2011 [15] and the SuperCollider Symposium in London, 2012 respectively. In May 2012 a related and more experimental work was performed, *Fluxus Tree*, which utilises the movements and touch of dancers to generate electroacoustic sounds and live notation for the instrumentalist, in this case the composer and (occasional, but excellent) 'cellist, Cheryl Frances-Hoad [16]. Comments and suggestions made by these excellent musicians have been vital in the development of these compositions.

Although I imagined the display of notation on a laptop screen would involve a fairly simple transposition of notation from paper to screen, the process actually revealed quite a number of issues and challenges for me and for those who have performed using the system. These became even clearer during the subsequent (iterative) processes of rehearsal, further development 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 so fully available to the performer of *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, and sometimes a phrase is not completed before a performer is required to play its opening notes.

Problems do arise: the performer may feel that they have been 'relegated to being a mere sight-reader (or expert improviser)' as one reviewer put it [25]. Alternatively, there are also 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. To encourage this I have tried to ensure that the music generated by the software is either not too difficult technically, or is music that can be 'improvised' with some ease (swirling chromatic passages, for example). Both Mifune and Cheryl have indicated, in response to the above criticism, that rather than being 'relegated' to sight-reading, they find the experience exhilarating [11, 23]. Mifune also confirmed that, as a typical performer of western contemporary music, she found it easier to improvise when provided with a notated basis.

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?

Another issue is when to actually display whatever form of notation is required, and in particular how much 'notice' of an upcoming event should be given to the performer. This isn't a matter of rehearsal but instead of how much time is required for someone to take in and react to the appearance of a musical phrase. Inevitably it can initially create nervous tension for the performer: often concerning how 'correctly' they are going to be able to play the appearing material. This reaction can be somewhat ameliorated when the performer becomes

aware of the inevitably improvisatory nature of some aspects of the performance. It also influences how the performer is to know when to play the notation in terms of measure and beat. I will consider this further in section 6.4 below.

In general it would appear 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 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 becomes 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 re-scheduled 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 highlights 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 challenge of an un-fixed score. Does it matter when the data is generated?

The delays mentioned above seem to be confirmed experimentally by research undertaken into musical sight-reading, where it is estimated that performers 'read forward' between 0.3 and 1.3 seconds of material [12, 21]. This is regardless of the number of notes to be read, although it is affected by their stylistic complexity. It should also be considered that the circumstances presented in *Calder's Violin* are not entirely the same: in 'standard' sight-reading the material does not appear and disappear effectively instantaneously. In any case, it indicates that a delay of about two seconds should be ample in most cases.

Why not, then, pre-render the notation to a fixed-media format and simply play it back? This would certainly make the process less 'dangerous' technically and save me a few nerves! There is one particular reason for this: the programming 'places' the 'printing' of the music to the screen at particular times and as such, as is described above, is itself an important aspect in determining when the music is itself performed (usually and depending on context, about a second and a half after display). In my original plans for the piece there was another reason: the influence of real-time sensor data on the music and

notation. This was not implemented in *Calder's Violin* but more recently has been in *Fluxus Tree* [16].

6.4. Other issues and questions

Another issue is whether to include visualisations of the accompanying synthesised piano material in order to help with coordination and prediction, a matter that increases in importance with the addition of further instrumental parts, but for *Calder's Violin* it proved unnecessary.

A ramification of not including such visualisations 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 durational algorithms that are designed to display material in clear and expressive ways, but of course duration is only one factor in the perception of tempo. 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. In rehearsal the performers seem to respond naturally to these inconsistencies in a way that makes me feel confident about the intuitive methods I have used in describing the notation's inherent vagaries. Of particular importance are the subjective interpretations of relationships between pulse and tempi; in conversation Mifune and Cheryl each confirmed that getting used to this new environment was indeed a matter of practice. What initially seemed vague and a source of anxiety became, with familiarity, liberating [11, 23]. Of course, this does not entirely resolve issues involving coordination and the piece as it stands takes some advantage of certain contemporary styles in only requiring precise coordination at a couple of carefully prepared moments (see figure 5 and 3'05" in Mifune's performance, 3' 28" in Marcus's) [15].



play together!

Figure 5. a moment of coordination with verbal instruction.

A final point regards common *annotations*, such as dynamic and phrasing marks. 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'. Eventually, however, it is intended that most traditional annotations should be available.

7. CONCLUSIONS AND FUTURE WORK

One of the most significant issues arising from this work regards timing, both in terms of scheduling delays and also in terms 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 in development, which should help to resolve this issue at least in part.

A related matter is how much, if any, 'history' (algorithmic data) to preserve, and how much access to it should be encouraged or even allowed. In other contexts it may be that the accumulation of historical data from sessions is 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? Responses from performers and organizers have proved a little contradictory, but in general seem to favour public display of the notation. While I'm happy with this, I am made a little uncomfortable by the thought that any reliance on an external visual display detracts from the independence and power of the music itself.

How much a part of the creative process could these technologies become? With easy integration of electroacoustic, notation-based, algorithmic and machine-listening realms potentially unified, many new approaches become feasible, even those that arise from behaviours that currently might be considered anomalous.

I am personally interested in integration of physical gesture with live notation: there are roles for this 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. I expect to make this publically available during the summer of 2012 [17].

8. REFERENCES

- [1] Agostini, A, and Ghisi, D., *Bach Project*, [Online], available at: http://www.bachproject.net/bach/home_page.html, [retrieved 27 January 2012], 2012.
- [2] Ariza, C. "The Interrogator as Critic: The Turing Test and the Evaluation of Generative

- Music Systems." *Computer Music Journal* 33(2), 2009.
- [3] Chadabe, J. Electronic music and life. *Organised Sound*: 9, p3-6, 2004
- [4] Collins, N. "The Analysis of Generative Music Programs." *Organised Sound* 13(3). 2008
- [5] Collins, N. "Musical Form and Algorithmic Composition." *Contemporary Music Review* 28(1): 103-114, 2009.
- [6] Didkovsky, N., & Hajdu, G., "MaxScore: music notation in Max/MSP". ICMC, SARC, Belfast. 2008.
- [7] Fober, D., Orlarey, Y., Letz, S., "INScore: An Environment for the Design of Live Music Scores". *New Interfaces for Musical Expression, Oslo, Norway*. 2011.
- [8] Freeman, J. "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance." *Computer Music Journal* 32(3): 25-41. 2008.
- [9] Freeman, J. & Colella, A. "Tools for Real-Time Music Notation", *Contemporary Music Review*, 29:1, 101-113. 2010.
- [10] Frances-Hoad, C., Hall, T., Hoadley, R., Turner, J. *Triggered*. Cambridge and London: Music, dance, hardware and software design, 2011.
- [11] Frances-Hoad, C. *personal communication*, 2012.
- [12] Furneaux, S. and Land, M. "The Effects of Skill on the Eye-Hand Span during Musical Sight-Reading." *Proceedings: Biological Sciences* 266:1436, 2435-2440. 1999.
- [13] Hajdu, G., Niggemann, K., Siska, A. & Andrea Szigetvari. "Notation in the Context of Quintet.net Projects", *Contemporary Music Review*, 29:1, 39-53. 2010
- [14] Hedelin, F. "Formalising Form: An Alternative Approach To Algorithmic Composition." *Organised Sound* 13(3): 249-257, 2008.
- [15] Hoadley, R., *Calder's Violin*. [Online], available at: <http://rhoadley.net/video/calder>, [retrieved 25 May 2012], 2012.
- [16] Hoadley, R., *Fluxus Tree*. [Online], available at: <http://rhoadley.net/fluxustree>, [retrieved 31 May 2012], 2012.
- [17] Hoadley, R., INScore class for SuperCollider. [Online], to be available at: <http://rhoadley.net/software/inscore>, [retrieved 31 May 2012], 2012.
- [18] Hudak, P., Makevich, T., Gadde, S., Whong, B. "Haskore Music Notation - An Algebra of Music." *Journal of Functional Programming* 1(1). 1993.
- [19] Loy, D. G., *Musimathics : the mathematical foundations of music*, Cambridge, Mass. ; London: MIT Press. 2006.
- [20] McClelland, C., Alcorn, M., "Exploring new composer/performer interactions using real-time notation". *ICMC, SARC, Belfast*. 2008.
- [21] Rayner, K. and Pollatsek, A. "Eye Movements, the Eye-Hand Span, and the Perceptual Span during Sight-Reading of Music". *Current Directions in Psychological Science*. 6(2). 49-53. 1997
- [22] Rowe, R., *Machine musicianship*, Cambridge, Mass.: MIT Press. 2001.
- [23] Tsuji, M., *Personal communication*, 2012.
- [24] Wulfson, H., Barrett, G., Winter, M., "Automatic Notation Generators". *New Interfaces for Musical Expression, New York*, 2007.
- [25] Unknown reviewer, *personal communication*, 2012.