



Hall, C., O'Donnell, J. , and Bailey, N. (2011) BowScribe: Supporting the violinist's performance model. In: International Symposium on Performance Science (ISPS 2011), August 2011, Toronto.

<http://eprints.gla.ac.uk/68408/>

Deposited on: 21th August 2012

BowScribe: Supporting the violinist's performance model

Cordelia Hall¹, John T. O'Donnell¹, and Nicholas Bailey²

¹ School of Computing Science, University of Glasgow, UK

² Centre for Music Technology, University of Glasgow, UK

Musicians often learn about their vision of a piece through practicing it and listening to recordings. However, this does not always free the player to develop his or her own interpretation of the piece, especially when technique is lacking. We have developed software, the *BowScribe* markup language, that supports a violinist in creating a “performance model” of a piece currently beyond his or her playing skills, by allowing the player fine control over tempo, volume, and articulation, including playing of chords, at a level of expressiveness and flexibility that is significantly beyond the MIDI playback modes of popular music notation software. BowScribe has been used by the first author (who was trained as a professional violinist) to create a model of the entire Bach *Chaconne* (edited by Galamian), a long and demanding piece of music for solo violin that has many phrases that span groups of chords as well as melodic passages. The markup language specified chords to be rolled in two classic ways, as well as a wide variety of other strokes, including greater volume for individual notes in long slurs and small but essential variations in tempo.

Keywords: violin; performance; tool; computer; model

String performers spend much time developing their technique. However, unless they grow as interpreters of music, their improved technique will not be sufficient to attain their musical goals. Our thesis is that it is possible to develop a software system that helps a string player to develop their ideal performance model of a piece, using annotations to express fine control over the performance.

There has been significant work done on synthesizing performances using rule-based systems (Frieberg *et al.* 2006) and tuning rule-based systems to

create a performance that is similar to that of a particular performer (Zanon and de Poli 2003). These lines of work are significant for our research because they demonstrate that performers have specific styles that are different enough to be worth studying. This is explicitly supported by work on automatically recognizing violinists playing Bach's solo sonatas and partitas (Molina-Solana *et al.* 2008). This has some interesting implications for the student who develops his or her own model. Such a student must abstract away from performances by prominent musicians, or absorb their habits and expressive characteristics. Other work on the importance of learning from a musical model (Rosenthal 1984) and taking the experience of the listener into account when evaluating the success of a performance (Gabrielsson and Juslin 1996) suggests that students constructing their own models would do well to involve the critical ear of others.

MAIN CONTRIBUTION

Our thesis is that software tools can help novice and intermediate players to work on their interpretation concurrently with technical mastery. To demonstrate this, we have developed a combination of notation and computer software that helps the player to create a "performance model" of the piece. The system is called *BowScribe*, and it has been implemented. The system takes music text expressed in the standard Lilypond format. The user then annotates the text file and runs our software application, which outputs a typeset score with the annotations, as well as a MIDI file enhanced by those annotations. The user can listen to the system's MIDI output, analyze it, and refine the annotations. Eventually, this process produces a performance model with which the user is currently satisfied, at which point the user can go back to developing the technique required to achieve the desired interpretation. The *BowScribe* notation and software are open source.

We have evaluated *BowScribe* by using it to develop a performer's model of some of Bach's solo works for the violin. Some existing software, such as the commercial Sibelius package, supports simple rolled chords, but more control is needed. In particular, we need a way to stop the roll at the point where the melodic line cuts through the chord. This can occur at the bottom, middle, or top of the chord. Sibelius and Finale support crude tempo changes in playback, indicated by metronome markings or tempo change notation such as *ritardando* etc. However, what is often needed is just a slight tempo change to express rubato on one or two notes within a whole phrase without rubato. Digital audio workstations provide powerful facilities for enhancing MIDI once it is generated by another tool, as does Rosegarden, an open

source audio and MIDI sequencer and processor. However, these are designed to treat MIDI just as a stream to be converted to audio.

An initial example

To illustrate BowScribe, consider a simple example. Figure 1 shows four annotated passages from the Bach *Chaconne*. The annotations were added by the violinist while working out an interpretation. The score shown in the figure was typeset by Lilypond, and BowScribe has generated a corresponding MIDI file. The first line gives bars 9 and 10, the second bar 30, the third bar 98, and the fourth bar 122. At the tops of these bars, there are annotations that are combined by our software with MIDI generated from the Lilypond text file.

We start our discussion with the bar 30 (the second line in Figure 1). All annotations start with “\$.” and the annotation for articulation is “*ln*”, where *n* ranges from short to medium (1 to 5), and medium to longer (6 through 9). The default sounds as if the note is slurred.

Looking at the first four sixteenth notes, we see that the first annotation will shorten a stroke with three notes. The second annotation gives a more precise way to identify the shortened note by specifying the part of the slur to be shortened, so for the second slur we write \$.x3.16, meaning that there are three notes which are to have the default articulation, followed by a note that will be shortened.

The figure displays four staves of musical notation for a solo violin part from the Bach *Chaconne*. Each staff is annotated with performance instructions. The first staff (bars 9-10) includes annotations like '\$b3 1', '\$.16', '\$b3 2', and '\$.b3.16'. The second staff (bar 30) has '\$.16', '\$.x3.16', and '\$.19'. The third staff (bar 98) features a series of '\$.17.i5' and '\$.i5' annotations. The fourth staff (bar 122) has '\$.c5' and '\$.ui7c3' annotations. The notation includes various note values, slurs, and dynamic markings.

Figure 1. Excerpt from the score of the Bach *Chaconne* for solo violin. Annotations have been added, as explained in the text. Our software generated MIDI incorporating the annotations, and Lilypond typeset the score shown here.

The annotation for relative note intensity is “*in*”, where *n* ranges from soft to medium (1 to 5) and medium to loud (6 to 9). This is useful when a note within a passage should be relatively loud.

For example, in bar 98 of the Bach *Chaconne*, we want to accentuate the notes at the top of the arpeggiated section and follow them as they descend. There are two annotations: \$.i7.i5 and \$.i5. The first indicates that the first of two notes will be louder than the second, whereas the second annotation returns the stroke to the default dynamics.

A more complicated example

The next example comes from the Bach *Chaconne* (bars 9 and 10), at the very beginning, and contains chords and bowstrokes of varying lengths (see the top line in Figure 1).

The new annotation we will need is one that indicates the type of chord used. If the notes in the chord are to be broken and one of the notes in the chord is played at the end, then we use “*bn*” where *n* indicates which note of the chord will be played last (1 for top string, 4 for bottom). If the notes of the chord are played simultaneously, or the upper string carries the melody, then we use “*u*” (which can also be used with single notes to economize on notation needed for passages in which single notes are mixed with chords that finish on the top note).

Notice that in this example several chords are rolled to the note on the top string and then to the bottom note. This is done so that the line of the melody, which is being played on the lower strings of the instrument, is emphasized.

Finally, we show an example (bar 122, the last line in Figure 1) in which a relative change in tempo is inserted. The annotation for relative change in tempo is “*cn*”, where *n* is slow to medium (1 to 5) and medium to fast (6 to 9). In the last line of Figure 1, we would like the first note of the bar to be a little longer and louder than usual, to emphasize the contrast between it and the rest of the bar.

Working with the Bach *Chaconne* as a whole

The Bach *Chaconne* is a piece for solo violin which takes 13 pages in the Galamian edition that we used. It is interesting to study because chords are pervasive throughout the piece and because it is composed of variations on a very small theme. In order to handle the chords in a satisfactory way, we must ensure that there is a way to emphasize the melodic line when it runs through a series of chords and a way to specify the articulation of the bow strokes playing the chords. The variations on the theme require careful study because

many of them continue on into the next variation without any change in phrasing, but some require substantial *ritardandi*, with emphasis on certain notes.

For example, in bars 1 to 8, the main theme appears in chords, with the melodic line at the top of the chords, and each bow stroke is long, allowing the chords to ring out without stopping the string. We use the annotation `$.ul7c4`, indicating that the articulation is longer and the tempo is slightly slower than usual. However, the next section, bars 9 through 16, is significantly different because Galamian indicates that the articulations of the strokes are shorter, almost like a dance, and the melodic line is at the bottom. Here, we use the annotation `$.b3.l6`, which does not specify a tempo change (thus resetting the tempo from the previous annotation, `c4`), and which indicates that the melodic line is carried at the bottom of the chord. Varied articulation annotations are used to heighten the interest during the rest of the section before bar 89, at which point the first arpeggio section starts.

Here, we emphasize the notes at either the bottom or the top of the arpeggios, and after some parts of that section are completed, we vary the tempo slightly. At the end of the section, the tempo should slow significantly to prepare for the later re-entry of the theme, again in chords. The rest of the piece is annotated in a similar way.

IMPLICATIONS

The BowScribe notation allows a violinist to express subtle aspects of the performance, and the software tools typeset the music and generate MIDI output. The system allows a string player to experiment with various choices in interpretation, and to listen to the result, even before their technique has reached the level required to play that interpretation. The system also makes it possible to achieve greater control over audio generated from MIDI.

As can be seen from our examples, writing Lilypond and annotations can be a job that requires precision. The current system demonstrates the capabilities of such a detailed notation and software tools, but it is not easy to use. In future work, we plan to experiment with an interactive WYSIWYG music editor that automatically generates the textual notations, while allowing the performer to use annotations that are as intuitive and expressive as possible.

Address for correspondence

John O'Donnell, School of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK; *Email*: john.odonnell@glasgow.ac.uk

References

- Friberg A., Breslin R., and Sundberg J. (2006). Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology*, 2, pp. 145-161.
- Gabrielsson A. and Juslin P. (1996). Emotional expression in music performance: Between the performer's intention and the listener's experience. *Psychology of Music*, 24, pp. 68-91.
- Molina-Solana M., Arcos J., and Gomez E. (2008). Using expressive trends for identifying violin performers, *Proceedings of ISMIR 2008* (pp. 495-500). Philadelphia: International Society for Music Information Retrieval.
- Rosenthal R.K. (1984). The relative effects of guided model, model only, guide only and practice only treatments on the accuracy of advanced instrumentalists' musical performance. *Journal of Research in Music Education*, 32, pp. 265-273.
- Zanon P. and de Poli G. (2003). Estimations of parameters in rule systems for expressive rendering of musical performance. *Computer Music Journal*, 27, pp. 29-46.