NETWORKED COMPROVISATION STRATEGIES WITH ZSCORE

Slavko Zagorac
Goldsmiths College, University Of London
`slavko@zagorac.com`

Michael Zbyszynski
Goldsmiths College, University Of London
`M.Zbyszynski@gold.ac.uk`

ABSTRACT

ZScore is a networked notation system for mixed ensemble composition and performance. This paper describes recent system developments towards a platform for comprovised music making. The long-term project objective is to provide an inclusive, democratised music-making environment by utilising technology that enables distributed decision making, dynamic notation processing and visualisation. It is proposed that all music is the outcome of a decision-making process that can be represented on the spectrum between immutable static and real-time dynamic decision making. Networking technology can act as an enabler for moving the dial on this decision-making spectrum in a required direction. Furthermore, a definition of a networked notational perspective is outlined, covering the dynamic aspects of distributed notation for heterogeneous clients. Several strategies for dealing with dynamic notation visualisation and control are presented, such as the dynamic performance parameter processing and embedded scripting. Finally, this paper presents the results from recent user trials and plans for future developments.

1. INTRODUCTION

Composition and improvisation are often regarded as mutually exclusive music-making categories. In practice, however, it is not possible to define a clear boundary between composed and improvised. Bhagwati [1] argues that no score can totally determine all aspects of a musical performance and that some elements of music-making will always be contingent. Likewise, a performer's free improvisation is built on years of practice and performance, stemming from a particular tradition and aesthetic context. A free improviser also adheres to a set of rules and regulations that might be imperceptible to a performer. It follows that any music performance lies somewhere on the spectrum between composed and improvised. The portmanteau word 'comprovisation' is often used to describe this mix. This paper considers music-making strategies that blur traditional boundaries and intentionally occupy the middle range of the comprovisation spectrum.

Music notations developed in different musical traditions tend to optimise the amount of written information to what is regarded as essential, and omit performance elements that are ubiquitous, verbose or too difficult to notate [2]. Traditions rely on the performer's understanding of the style, playing techniques and aesthetics to complete required information gaps missing in the score. In the contemporary context, this notational bias, described as 'notational perspective' [2], has a more granular scope that does not only define a tradition or a style of music, but helps identify individual comprovisation practices. Networked notation technology further extends this concept by allowing for the creation of dynamic notational perspectives that can be modified in real-time. The dynamic notation, in this case, becomes a context-dependent contingent element in a performance. This notation contingency can be achieved through generative symbolic notation algorithms or dynamic performance parameter control described further in this paper.

Existing networked comprovisation software solutions typically consist of a front end capable of rendering dynamic notation such as: InScore [3], MaxScore [4], Bach [5], internet browser etc.; and a server engine in charge of notation scheduling and distribution, usually hosted in software containers such as MaxMSP, Processing, Node.js or programming language environments like SuperCollider and ChucK. ZScore [6] is a networked notation system providing real-time notation distribution and performance flow control. It utilises InScore for notation visualisation and a custom-built Java engine for algorithmic notation processing and distribution. This paper outlines ZScore comprovisation strategies, focusing on a dynamic notational perspective and real-time decision-making controls.

2. COMPROVISATION DECISION-MAKING SPECTRUM

A piece of music, whether composed or improvised, is conceived through a decision-making process defining what sound or action is to be performed and at what time. In a composed piece, most music material decisions are made pre-performance in isolation by a composer who preserves these decisions as notation realised in a static score. Even in the most meticulously notated scores, however, many performance decisions are left to the performers who interpret the given notation based on their experience and knowledge of the particular music tradition. Rodrigo Constanzo [7] developed a formal methodology for analysis of the decision-making process in improvised works. His segmentation of the music-making decisions into material, formal, interface and interaction, illustrates the improviser's real-time dynamic decision-making approach in an interactive group performance environment. It could be pos-

Copyright: ©2020 Slavko Zagorac et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

tulated that all music-making exists within the comprovisation decision-making spectrum between predetermined statically notated decisions and dynamically made decisions in real-time (Figure 1). An important consideration

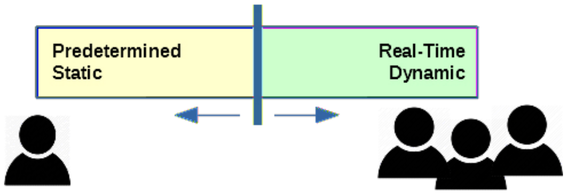

Figure 1. Music performance decision-making spectrum.

when discussing the comprovisation decision-making spectrum is to observe who the decision makers are, the type of the decisions made and their impact on the aesthetic evaluation of the piece of music. Traditionally, composers' decisions are immutable. If a performer intentionally or accidentally modifies a statically notated piece of music during a performance, it is generally interpreted as the aesthetically unsatisfactory realisation of a piece. In a free improvisation performance, performers are allowed to make individual decisions at any time. Although there are no right or wrong decisions in this case, the aesthetic value of a performer's decision is continually assessed within the context of a particular performance by the audience, the performer who made the decision, and by all other performers who react in a chain of decisions resulting in a self-regulating musical output. Comprovisation pieces that sit in the middle range of the decision-making spectrum contain both static immutable elements and a dynamic decision loop created by multiple decision makers in real-time. Concepts such as static data containers, networking, interactivity and real-time dynamic event processing are ubiquitous in modern computer technology, therefore, technology is well placed for modelling and enhancements of comprovised music-making data and decision processes. The intention behind ZScore is to provide a platform for the democratisation of music-making processes and unconstrained control of the comprovisation decision-making dial (Figure 1) as required by the context of a performed piece of music.

3. NETWORKED COMPROVISATION NOTATIONAL PERSPECTIVE

Conventionally, a music score consists of notation acting as a set of instructions or stimuli required for the realisation of a piece of music as imagined by the composer. As discussed in section 1 (Introduction), all scores contain a certain bias regarding the type and quantity of information defined as a 'notational perspective'. The quantity of information in a score should reflect the composer's intention as to what the piece of music is within the assumed notational perspective. 'New Complexity' composers [8] create dense, detailed, deterministic scores that intentionally challenge boundaries of performability. Jazz music notation often consists of a melody (a tune) and related chord progression charts, leaving out the instrumentation, dynamics and even tempo. Many composers utilise elements of indeterminacy in their compositions (Cage, Brown, Lutosławski etc.) and their scores are constructed to reflect intended contingency, thereby frequently containing graphic or unusual symbolic notation layouts. In all cases, the score performance evaluation should examine whether the musical outcome matches the composer's intention. Networked notation systems allow heterogeneous clients to connect and exchange data and events over a computer network. In such systems, a music score can be modelled as a collection of data and algorithms driven by a set of scheduled or triggered events. All participants in a networked music performance (musicians, composer, conductor, audience, digital audio and video engines, etc.) can communicate with each other through an interactive system in real-time. This multidirectional flow of information blurs the boundaries between traditional music-making roles, as any participant in a performance can be assigned a decision-making or a sound production agency. In a way, networked systems provide an environment reminiscent of ancient communal music-making (also described as 'musicking' [9]). Audience members, as networked system participants, can actively engage in a performance through personal mobile devices. The composer and musicians can interact in real-time during a performance to modify the composition flow, a conductor can change generative algorithm parameters to alter notation sent to the performers, and a computer running an algorithmic engine can trigger score visualisation on audience mobile devices etc.

This heterogeneous client environment requires multiple score representations as each network client might require a different input type. For example, a score on musicians' devices might be rendered as a symbolic notation while the same score on the audience's devices might be visualised as animated graphics. In Simon Katan's *Conditional Love* [1], the audience's mobile devices are used both for score visualisation and as audio sources. The audience's score representation can also contain a form of instructional notation, prompting the audience to perform certain actions at specific points in time, e.g. vocalise certain words. Furthermore, the audience's score representation on mobile devices can contain interactive elements, thus allowing the audience to send real-time events to other participants via an algorithmic notation engine. A part of the same score might be sent to a digital video engine translated to a device-dependent data protocol, as in Slavko Zagorac's *Vexilla* (Figure 7). Likewise, the score data can be used to drive audio engines implemented in MaxMSP or SuperCollider.

Composers have to consider several new performance aspects in this interactive networked environment. The score becomes more than just a static notation collection. It requires a definition of dynamic performance elements and a strategy on how to present and deal with real-time events. The dynamic actions display should not interfere with the rest of the notation on musicians' screens. Performers need time to perceive any actionable elements and consequently take any required actions. Therefore, reaction time needs

[1] S. Katan. *Conditional Love*, last accessed 12 Jan 2020, `http://www.simonkatan.co.uk/projects/conditionallove.html`

to be built into the score in a way that does not impact the performance flow. All participants need some kind of confirmation that their actions have been processed. If it is not immediately audible what the outcome of their actions is during a performance, then some kind of a visual acknowledgement is required. Likely issues with the network technology during a performance need to be taken into account. Some kind of default outcome should be defined if real-time events cannot be processed due to technical issues. The aesthetics of networked comprovisation will be evaluated through the participants' ability to perceive the score realisation and whether the impact of their decisions matches their desired musical output.

4. DYNAMIC NOTATION STRATEGIES IN ZSCORE

Unlike static notation, networked dynamic notation views for instrumental parts require a carefully thought-out refresh strategy that does not interfere with any currently played notation, whilst leaving enough look-ahead time for the performers to prepare for the upcoming material. The alternating pane notation strategy in ZScore [6] aims to resolve dynamic notation update issues by providing familiar left-to-right and top-to-bottom reading directions and pre-defined time windows when upcoming notation can be updated, thus leaving ample look-ahead time for performers (Figure 2). At any point of time during a performance, there is always one active and one preparatory pane. The preparatory pane is updated with upcoming notation within the pre-defined time window while the active pane is played [6]. Currently, ZScore notation is a com-

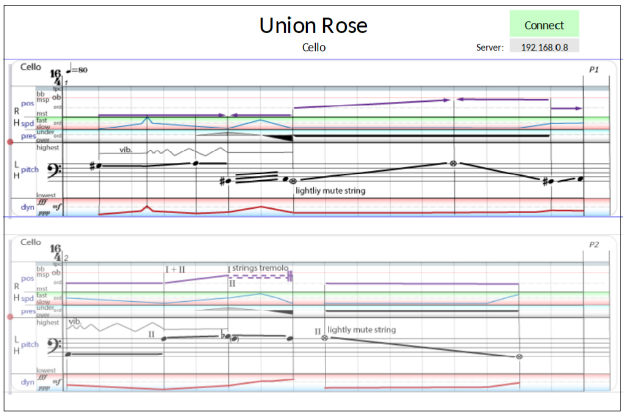

Figure 2. Cello part from *Union Rose* by Slavko Zagorac.

bination of static files generated in Adobe Illustrator and dynamic SVG (scalable vector graphics) content rendered by InScore. This approach allows for syntax-independent and constraint-free positioning of notation elements suitable for various dynamic notation styles. Following the usability first principle, however, the current design preference is to provide a consistent layout familiar to classically trained musicians. Early versions of the notation layout used a mixed symbolic/graphic notation space, offering optimal and flexible screen real estate utilisation (Figure 7). However, after the feedback received through research questionnaires and verbal conversation, it became clear that the majority of musicians preferred distinct and consistent positioning of different performance parameters rather than the mixed space approach. Furthermore, some notation element positioning orders felt more 'natural' to performers than others, such as having the dynamic markings always displayed below the pitch information. The above considerations and the multiple user trials have led to the current layout design illustrated in Figure 2.

4.1 Dynamic Performance Parameter Notation

Performers intuitively apply complex playing techniques when reading a notated score, based on years of learning, practice and performance. Classically trained string instrument players, for example, automatically translate music dynamic markings into multiple bowing techniques, such as the control of the bow speed and pressure. When interpreting symbolic notation, string players strive to produce a sound quality that satisfies aesthetic requirements of the particular tradition or style. This anticipation of the required 'ideal' sound then translates into learned application of the bowing techniques. The artistic aim in the latest pieces written for ZScore is to create a particular sound quality through notation that intentionally decouples learned mapping between the notation, playing techniques and sound. Additionally, the objective is to have a system suitable for both static and dynamic notation that allows for a flexible control of different playing techniques in real-time.

User trials and various concerns described above lead to the notation layout where the key performance parameters are separated into distinct two-dimensional Cartesian coordinate spaces. Figure 3 illustrates a vertical stave layout for string instruments taken from the score shown in Figure 2. The stave is vertically split into three main sections

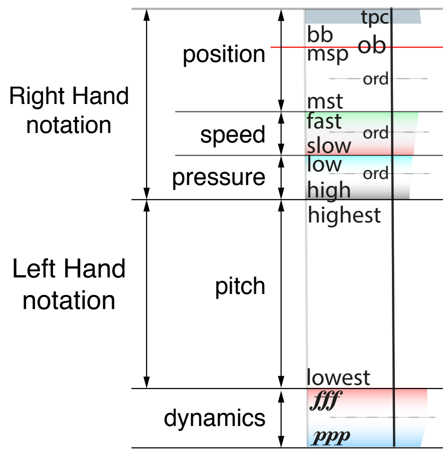

Figure 3. String instruments vertical notation layout.

dedicated to the right hand, left hand and music dynamic notation. The right hand notation section is split into three subsections: position, speed and pressure notation. These subsections refer mostly to the bowing technique but can contain other actions such as pizzicato or percussive sound gestures. Left hand notation contains pitch or a playing technique information in either symbolic or graphic notation. The dynamics section is always displayed below the pitch information based on the feedback from the musicians during user trials. The vertical axis in each section (Figure 3) represents a named performance parameter value. The vertical range is parameter specific. For example, bow position range for all string instruments starts with molto sul tasto (mst) then continues with markers for ordinary playing position (relative to the given dynamics), molto sul pont (msp), on bridge (ob), behind bridge (bb) and ends with the tailpiece (tp) marker. The horizontal axis represents time and is identical for all performance parameters. The position cursor rendered as the green line across the stave shows the current position on each stave at any point of time (Figure 4). Additionally, the dynamic beat cursor in the shape of the red bouncing ball indicates the onset of each beat, similar to a conducting gesture. The

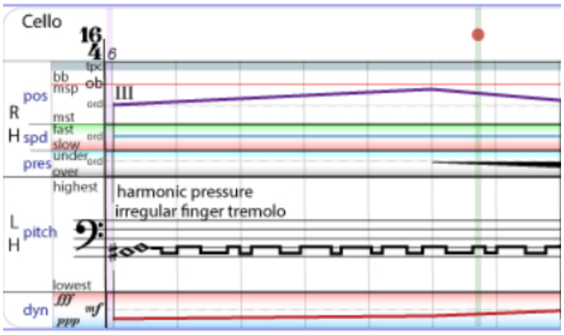

Figure 4. Cello part notation excerpt from *Union Rose*.

cello part excerpt (Figure 4) taken from *Union Rose* score (Figure 2) illustrates a pre-composed static notation layout. The current value for several performance parameters (position, speed and dynamic) is visualised as a continuous line. Pressure notation uses coloured geometrical shapes indicating the amount of bow pressure that needs to be applied relative to the current dynamics. Left hand notation, in this case, is a combination of textual, symbolic and graphic notation indicating finger pressure, position and timing. The finger tremolo timing information is an approximation that illustrates an idea of an irregular finger placement. It is left to a performer to decide the exact timing by using their aesthetic judgement at the time of the execution. This decision might be different every time the section is played as the overall context and the surrounding sound output can change with each pass as described below.

Real-time dynamic decision making and notation rendering require a user interface design that allows for dynamic elements to update at any point of time without any detrimental impact on the displayed static notation elements and musicians' look-ahead preparation time. Dynamic notation updates need to be clear, easily understandable and suitable for real-time cognitive processing. Therefore, complex notation updates should preferably be scheduled for display in predetermined time window slots as described above. As a general guideline, the performance continuity should not suffer at any point on dynamic notation update, unless it is an intentional side effect. One of the reasons that lead to the performance parameter notation separation was a need to provide dynamic parameter value overrides. In order to achieve dynamic overrides, each parameter is assigned a graphic overlay [2]. Overlays sit on top of the

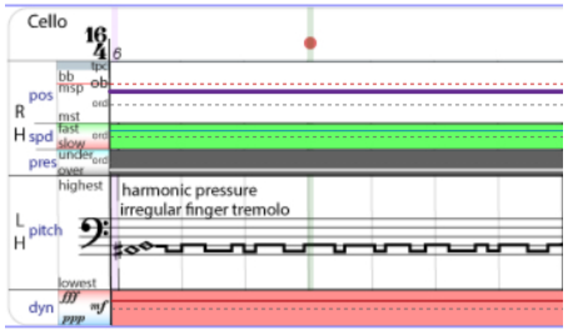

Figure 5. Dynamic notation overlays for position, speed, pressure and dynamics.

static notation covering the entire Cartesian space assigned to the corresponding performance parameter. The current dynamic parameter value is rendered either as a horizontal line on top of the overlay and/or as an overlay background colour value. Figure 5 illustrates the position, speed, pressure and music dynamic overlays for the same notation excerpt shown in Figure 4. In this case, only the pitch information remained the same as the pre-composed static notation. As can be observed in Figure 5, each parameter's current value is represented by the coloured line (red for dynamics, purple for position, blue for speed, grey for pressure) and the background colour covering the entire two-dimensional Cartesian space assigned to the parameter. In this instance the player is asked to play forte sul pont with fast bow and strong overpressure. Bow speed and position markings are always relative to the indicated music dynamics. Parameter values are controlled from the ZScore control GUI described below and have a preset range (min = 0, max = 100). The parameter's line position is obtained by

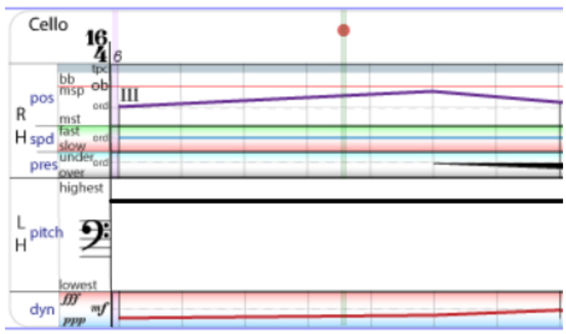

Figure 6. Dynamic notation overlay for pitch only.

mapping the current parameter value to the vertical space assigned to the parameter (minY, maxY). Similarly, the

[2] S. Zagorac, "ZScore Dynamic Notation" video, last accessed: 30 Dec 2019. `https://youtu.be/Yh6wUqLZwkU`

background colour is interpolated from the range representing minimum, middle and maximum value. For example, the music dynamics colour is interpolated from the range min: blue[R=0, G=0, B=255], mid: White[R=255, G=255, B=255], max: Red[R=255, G=0, B=0]. Overlays and parameter values can be set independently at any point of time and are immediately displayed on the musicians' screens. Figure 6 illustrates a different overlay configuration for the same score excerpt as in Figures 4 and 5. Here, only the pitch parameter has been overridden with the overlay, indicating an approximate pitch to be played within the instruments range (min = lowest, max = highest possible pitch). All other performance parameters in this example are pre-composed. In a similar fashion, any overlay can be switched on or off during a performance.

4.2 Embedded Scripting

In ZScore, actionable commands can be embedded directly into the score, as illustrated in in the excerpt from *Vexilla* by Slavko Zagorac (Figure 7). Here, the AV (audio/video) part contains commands in the textual form. These commands are loaded together with the score and sent to the server for scheduling and distribution to the networked video or audio engines. In this instance, JavaScript and SVG commands are routed to an InScore client which renders a video signal displayed to the audience via a video projector. The AV command scheduling resolution in this example is one bar, so the commands are attached to a bar container and scheduled to be executed on the onset of the starting beat of the bar. The commands are executed immediately on the client side. Some commands have a duration, such as a transition between two colours, and are executed over a period of time based on command settings.

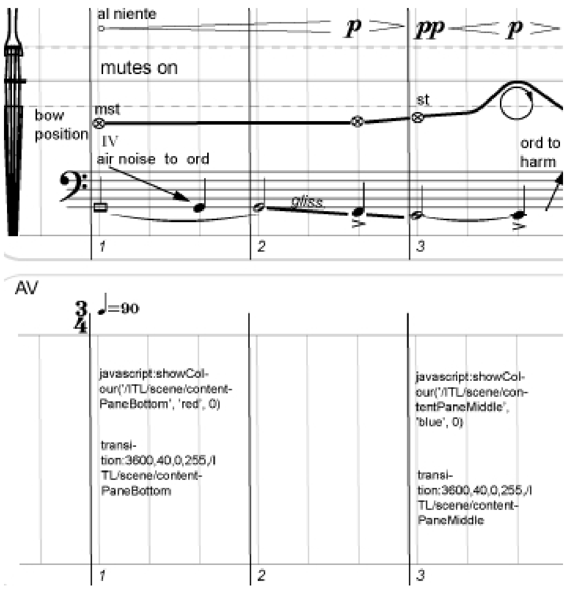

Figure 7. Embedded scripts in AV part from *Vexilla*.

5. ZSCORE PERFORMANCE CONTROLS

Dynamic performance parameters are controlled from the ZScore's administration GUI, written in Java FX (Figure 8). Currently, this GUI also allows for import of static score data, communication with performance participants and their connection monitoring, various algorithm control selections, as well as the score position and start/stop controls. Performance parameter overlays described in the

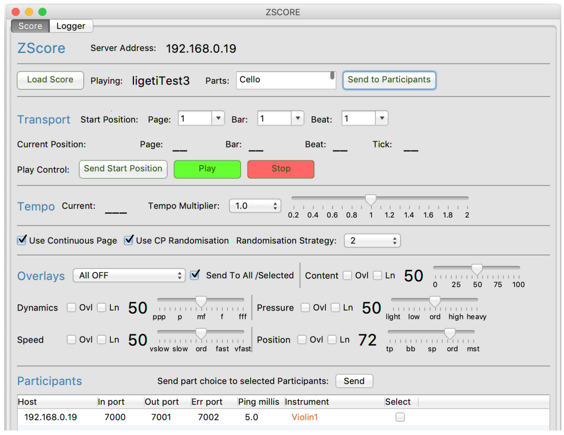

Figure 8. ZScore performance control front end.

previous chapter can be switched on or off at any point during a performance from the administration GUI. Here, the performance parameter ranges are displayed in musical terms. For example, the dynamics range is displayed as 'ppp - p - mf - f - fff' (Figure 8). The administration GUI communicates with other performance participants via ZScore's server, which provides notation scheduling and distribution service. The server also receives events sent by other performance participants, passes them through an algorithm which analyses incoming events and validates decision logic (Figure 9). The outcome of the decision logic is then passed to the score processing and scheduling engine which identifies required notation and distributes it back to the performance participants.

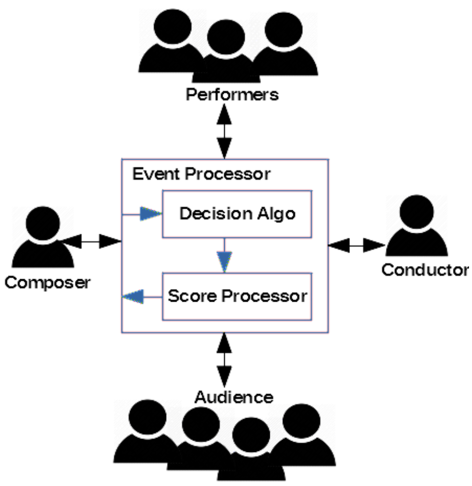

Figure 9. Decision event flow.

One of the the administration GUI's features is the randomisation strategy configuration selection. The randomisation algorithm decides the notation and instrumentation that should be used in the next time window as calculated by the scheduling engine. Figure 10 shows the available randomisation strategy configurations from *Union Rose* written for a string quartet. The configuration determines the number of distinct score pages and instruments that should be used. The term page here is equivalent to the notation displayed in a single pane. For example, the configuration value '2' means that two randomly selected instruments should play the same randomly selected page; the configuration value '2,1' means that two instruments should play the same page and one instrument should play another randomly selected page; '1,1,1,1' means that all four string quartet instruments should play a randomly selected page, and so on. *Union Rose* pages have been constructed to

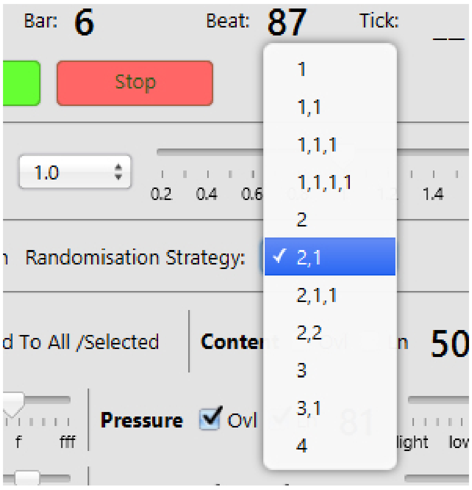

Figure 10. Randomisation strategy configuration control.

function as an independent unit, reusable in various vertical structure combinations. The aesthetic impact of sounding page combinations is evaluated in real-time during a performance. Any undesired sound elements can be modified with dynamic performance parameter controls to rebalance the page combination musical outcome.

6. COMPROVISATION USER TRIALS

ZScore has been used successfully in several live performances and workshops featuring networked score distribution, embedded scripting and synchronized multimedia. The latest version with the notation layout modifications and real-time comprovisation features described above was tested in a November 2019 workshop with the Ligeti Quartet [3], where an excerpt from *Union Rose* by Slavko Zagorac was performed [4]. The aim of the workshop was to validate the notation layout design changes and test technical stability and interactive aspects of the dynamic performance parameters.

The feedback from the musicians was mostly positive, citing: significant notation layout improvements compared to the previous versions; a 'natural' feel of the notation flow and the real-time performance parameter visualisation; a very good system stability; and accurate performance synchronization. The comprovisation features attracted a great deal of interest, including comments on how the repeated, familiar music material sounded completely different depending on the randomisation strategy configuration and dynamic performance parameter value selection. It was interesting to observe when and how musicians perceived dynamic notation changes. Initially, musicians focused almost entirely on the pitch notation section. Gradually, as they were getting more familiar with the composition material, they widened the scope first to dynamics notation and then eventually to the left hand performance parameter changes. Bowing pressure dynamic changes were only audible once musicians went several times through the same material and were comfortable with the pitch and dynamics notation. For future performances, the composition material will be distributed to musicians beforehand to give them some time to familiarise themselves with the piece. In addition, the performance plan will be modified to introduce dynamic performance changes gradually.

The musician's criticisms were mostly aimed at the quantity of visual markers, such as helper lines for mid-level performance parameter values, which increased the complexity of the user front end. The stave layout design is a compromise between notation precision and readability. In order to improve score readability, new features, such as current performance parameter value indicators (described in section 7: Future Work And Conclusions) will be implemented. Furthermore, the notation layout configuration features could be made available to musicians, thus allowing individual musicians to customise the notation layout as required.

7. FUTURE WORK AND CONCLUSIONS

At the moment all dynamic performance controls in ZScore are only available in the administration GUI. One of the ZScore project aims is to democratise the decision-making process and distribute performance controls to other participants. The short-term aim is to replace the randomisation strategy described above with separate controls for instrumentation and page selection. The musicians' front ends will be expanded with controls to allow for the selection of instrumentation, so in *Union Rose* for example, the quartet members themselves will be able to select instrument groupings and override any randomisation strategy instrumentation decisions. (Figure 11). Furthermore, the page selection and sequencing decision will be given to the audience. An example of the audience score visualisation for *Union Rose* by Slavko Zagorac is shown in Figure 12. The inspiration for this piece is the large stained glass rose window in Union Chapel, London, where the piece is scheduled to be performed. The audience score visualisation is a digital representation of the rose window split into 64 tiles. Each tile will be mapped to a score page which will allow the audience to select the path through the piece by clicking the next tile to be played on their mobile devices. The algorithm on the server side will process user input and schedule notation distribution as required. In order to improve the precision of dynamic per-

[3] November 2019 workshop with the Ligeti Quartet was funded by the Goldsmiths College Music Department

[4] S. Zagorac, "Comprovisation with ZScore" video, last accessed: 10 Jan 2020. `https://youtu.be/2pBqepq3Khc`

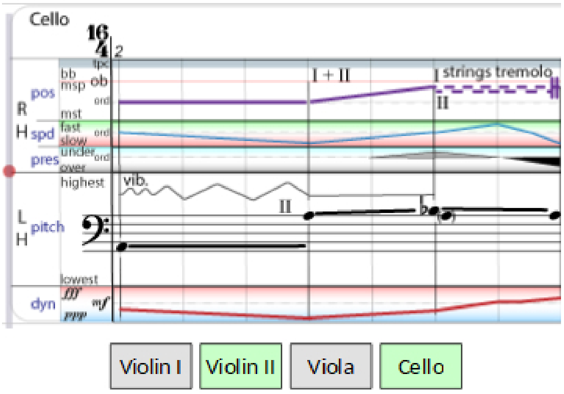

Figure 11. Musician's instrumentation selection controls.

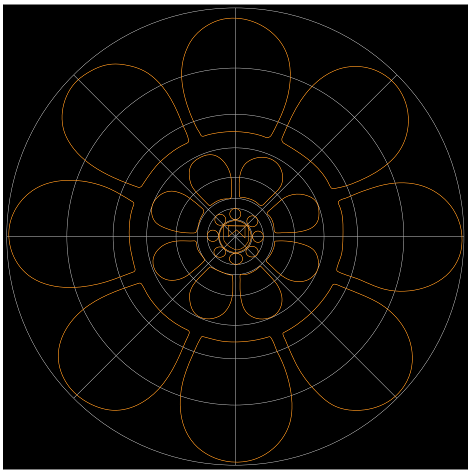

Figure 12. Audience score visualisation for *Union Rose*.

formance parameter display, an indicator of the current value will be attached to the current position tracker. An example of the music dynamics current value indicator is shown in Figure 13 ('mp' in the square box attached to the current position line). The indicator value will be set in real-time to reflect the dynamic parameter value. This should improve readability and decrease the notation layout complexity, as some markers, like the parameter mid-value lines, can be removed from the stave layout. The next

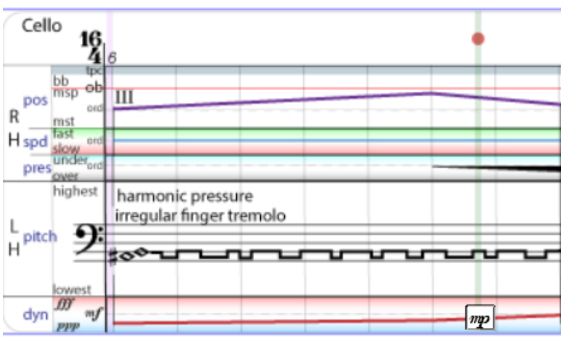

Figure 13. Current dynamics value indicator.

version of ZScore will also provide dynamic SVG notation positioning both for symbolic and graphic notation styles.

The score encoded in YAML format will allow for unconstrained named symbol positioning. Figure 14 shows two examples of symbolic notation, one where a named pitch is placed on the staff (G1) and one where a notehead (diamond) is explicitly placed in the x/y Cartesian space of the part's pitch notation area. Once the YAML score is loaded

```
notation:
  - { p: G1/4, stem: up, staff: 1}
  - { notehead: diamond, dx: 0.04, dy: 0.03 }
```

Figure 14. Symbol notation in YAML format.

into the ZScore sever engine, it could be passed into a generative algorithm capable of creating real-time notation for all participants. Symbol names (e.g., notehead=diamond) are mapped to SVG file names on the client side. SVG files representing all required notation symbols for a performed piece will be loaded on all notation clients prior to the performance so once the score symbol name is mapped to a file name it can be rendered immediately on the client side.

So far, several ZScore project goals have been achieved, such as the reliable real-time notation distribution over a network and dynamic performance parameter control, enabling networked comprovisation. The distribution of dynamic controls to different performance participants and multiple score visualisation representations are the next steps in the journey towards an inclusive comprovised music-making platform. This will require carefully thought-out strategies for real-time event management and score processing. The score composition process will have to evolve to take into account the impact of real-time events and their consequences. Contributions to these decision-making processes from both performers and audience members will inevitably bring additional risks and responsibilities to a comprovised music performance. The aesthetic evaluation of music-making in this shared comprovisation environment will, therefore, have to take into account the analysis of how well the musical and visual outcomes match the participants intentions. The hope is that this interactive human-computer system will lead to a new kind of musical and visual aesthetics.

8. REFERENCES

[1] S. Bhagwati, "Comprovisation – Concepts and techniques," *Re) Thinking Improvisation: Artistic explorations and conceptual writing*, pp. 99–103, 2013.

[2] ——, "Notational perspective and comprovisation," *Sound & Score. Essays on Sound, Score and Notation*, pp. 165–177, 2013.

[3] D. Fober, Y. Orlarey, and S. Letz, "Programming Interactive Music Scores with INScore," in *Proc. SMC Sound and Music Computing conference*, 2013, pp. 185–190.

[4] G. Hajdu and N. Didkovsky, "MaxScore - Current

State of the Art," in *International Computer Music Conference*, 2012, pp. 156–162.

[5] A. Agostini and D. Ghisi, "Bach: An environment for computer-aided composition in Max," in *International Computer Music Conference*, 2012, pp. 373–378.

[6] S. Zagorac and P. Alessandrini, "ZScore: A Distributed System For Integrated Mixed Music Composition and Performance," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, Montreal, Canada, 2018, pp. 62–70.

[7] R. Constanzo, "Making decisions in time," last accessed: 27 Dec 2019. [Online]. Available: http://www.rodrigoconstanzo.com/2015/04/making-decisions-in-time/

[8] S. P. Duncan, "Re-Complexifying the Function(s) of Notation in the Music of Brian Ferneyhough and the "New Complexity"," *Perspectives of New Music*, vol. 48, no. 1, pp. 136–172, 2010.

[9] C. Small, *Musicking: The meanings of performing and listening*. Wesleyan University Press, 1998.