COLLABORATIVE LIVE COMPOSITION WITH FRANKIE

Esa Onttonen
Sibelius Academy, University of the Arts
Helsinki, Finland
esa.onttonen@uniarts.fi

ABSTRACT

This paper describes a web application being developed for the purpose of collaborative live composition. In this concept, musical information such as pitch sets, chord progressions and textual parameters can be created and communicated in real-time by all members of the group. Primarily designed for tablets, the application aims for high usability values and has been tested and developed with professional jazz musicians. This paper presents the background for the concept of collaborative live composition, along with the main features, design principles and the development process of the application. Finally, the paper describes a concert performance and reflects on the affordances, opportunities, and challenges of the concept.

1. INTRODUCTION

Several methods to shape ensemble music in real-time have been introduced in the digital and non-digital domains. With their predefined signs and gestures, the conduction of Lawrence D. “Butch” Morris and the soundpainting of Walter Thompson are two better-known examples of non-digital methods, with Morris often crediting jazz drummer Charles Moffett as an important influence for his conduction technique [1, liner notes]. Having been described as group improvisation (conduction) and live composition (soundpainting) [2], they share some similarities with the rule-based game pieces by John Zorn, most notably *Cobra*, where the role of the prompter—usually played by Zorn himself [3]—is less about conducting and more about being a ”conduit of information” who responds to the requests of the musicians [4].

In the digital domain, various systems have been developed for real-time ensemble music making as well, such as John [5], Adaptive Markov Network for Free Improvisers [6], The Bucket System [7], live coding systems such as BEER [8], or more compositionally oriented systems such as ZScore [9] or Indra [10]. Unlike conduction, soundpainting, or Zorn’s game pieces, these systems usually do not have a conductor, unless the system itself or a generated score [5] is considered to be one. Additionally, the digital and non-digital examples presented have been mostly concerned with the music’s structure and the coordination of the structure, often seen as a challenge in ensemble improvisation [8, 11].

Although The Bucket System is implemented for QuNeo pad controllers without any visual notation, some of its fundamental ideas resonate with the project presented in this paper. They are both created as generic collaborative tools for professional musicians with the intention of allowing ”on-stage compositional/improvisation interaction”, and giving control to *all* musicians [7]. However, while The Bucket System selects randomly from the input given by the musicians, and John generates a random score based on constraints, Frankie does not employ randomization in the decision-making, but allows a single musician to be in control for any period of time until someone else takes the control, thus resembling alternating decision control of rotating leadership [12].

There are other differences to The Bucket System and the previously described digital improvisation systems as well. The improvisational aesthetics of the project presented in this paper are closer to Clifford or James than Earle Brown, often cited in the previous research [7, 5, 10], as well as by Zorn [3]. This has resulted in some features that are probably more useful for musicians coming from the jazz tradition. For example, improvisation in many subgenres of jazz is based on tonal and modal chord progressions and loops. Therefore, the application presented has been primarily developed for musicians who improvise on chord symbols and chord progressions, and pitch sets such as scales or melodic fragments, all available in the skill set of professional jazz musicians.

To define the concept of collaborative live composition, soundpainting provides a useful point of reference. In soundpainting, defined as *the* “universal multidisciplinary live composing sign language” [2], the composition is created by the soundpainter. In the project presented here, all members of the group can contribute to the directions the music will take, therefore it is defined here as collaborative live composition.

Furthermore, the live composition can be played *as such*, or it can provide a springboard for improvisation, as in jazz compositions. Although the term *comprovisation*—already employed by Butch Morris in the 1980s before settling on conduction [13]—is often used for compositional-improvisational practices especially outside jazz [14, 7, 5, 9], in this paper, the term composition is used with the jazz approach: a composition may consist only of a minimum amount of information—e.g. Carla Bley’s miniatures [15, p.16]—which is realized by the improvising musicians with a varying degree of connection to the source material.

The long-term development goal of Frankie is to create an easy-to-use tool that runs on a standard web browser on a typical mobile device and requires very little learning effort from musicians who are trained in Western common practice notation (CPN). Musicians can use their skills in sight-reading CPN which subsequently can provide triggers for their imagination [16]. As a communication tool for collaborative live composition, it does not come loaded with musical content or rules: it is up to the musicians to decide how to deal with the information communicated through the application. Additionally, the application does not try to replace any existing and perfectly valid cue mechanisms such as indicating a downbeat with an arm movement or a nod of the head; in fact, it does not currently provide any timing features. For its research part, the project aims to find out the musical affordances of this kind of practice by exploring the domain between composition and improvisation with the assistance of the application.

2. MAIN FEATURES

The following list describes the types of information that can be communicated with the current version of the application, including the descriptions of the interactions. All parameters, options, keywords, dynamics and roles can be freely configured in JSON (JavaScript Object Notation) files stored on the server.

- Short notated score fragments—simply called scores—of either durationless pitch sets (*set mode*) or chord progressions with automatic chord symbol recognition (*chord mode*). The score is entered with a MIDI input device and when complete, a swipe-up gesture on the score area sends the score to other musicians (alternatively, a button can be pushed). A short animated flash happens on the receiving devices to inform about the new score more visibly than simply changing the notation on the staff.
- Time signature, tempo and other custom parameters in drop-down menus that are displayed in the top section of the score area.
- Keywords as toggle buttons. Multiple keywords can be selected simultaneously, and they can be cleared with a dedicated button. These are displayed in the bottom section of the score area. The keywords are comparable with *karmas* of John [5] and to a lesser degree with the *signals* of The Bucket System [7].
- Dynamics which can be set individually or for all musicians at once from a drop-down menu. The dynamic mark is displayed in the lower left corner of the score area below the clef.
- Roles. Separate from the leader/musician division described later, each musician can have multiple simultaneous roles that are selected from a checkbox menu. For musicians their roles are displayed in the upper left corner of the score area above the clef. The roles can be used to define the musical function a musician should take. For example, a *bass* role indicates that a musician should treat the incoming material as a bass player, while *solo* could direct towards playing a solo. A combination of the previous roles, *bass* and *solo*, would suggest playing a solo in the bass register.
- Text messages that can be configured in advance or written during a performance.

By default, all interactions are transmitted immediately. For example, clicking a keyword sends it right away to the musicians. This works well when the objective is to change the music incrementally, one parameter at a time, for example from quiet to louder dynamics or from slow to a faster tempo. Sometimes it might be more suitable to build a more comprehensive content set where a score, parameters and keywords are all sent at once. This can be achieved with a hold-release functionality, where turning on a hold switch holds all communication until a release button is pressed. For a more fine-grained communication, it is possible to select the recipients and send different scores and parameters to different musicians.

In addition to the transmittable content described before, the application comes with some utility features. The musician who is the leader can turn on a blackout mode which dims all screens. This mode can be used for more traditional free improvisation, and it allows the musicians to take a break from following the screen. A score storage, another utility feature, allows the user to store up to 12 scores [1] with their parameters. Although this functionality is in some contradiction with the concept of *live* composition, it allows creating musical structures by revisiting previously played scores.

There are two user interfaces, leader and musician, depending on the status of the musician (see Figure 1). Although the design and the purpose of these interfaces are different, they share certain elements. The color scheme of the application is dark, so that the screen does not shine too brightly on dimly lit stages. In both interfaces, the main score area is displayed in the top area of the screen with most of the other elements placed below. While a tablet screen could contain even larger scores with more staves, the area has been deliberately kept compact to allow smaller devices such as mobile phones to be used for displaying the notation in reasonable size. Additionally, in the bottom bar there are some features that are common to the leader and musician interfaces (musician and session info, tutorial, reload, settings, full screen, clock), as well as some features that are available only in specific interfaces, such as the blackout mode in the leader interface or the 'Take Leadership' button of the musician interface.

2.1 Leader interface

The leader interface is used to compile a score and set of instructions for other musicians, and it is available for the musician who is currently in the leader role. The leader can

[1] The limitation of 12 scores is not technical but by design to allow using a MIDI input device to quickly send a score from the storage with any C note sending score 1, C# sending score 2 and so on.

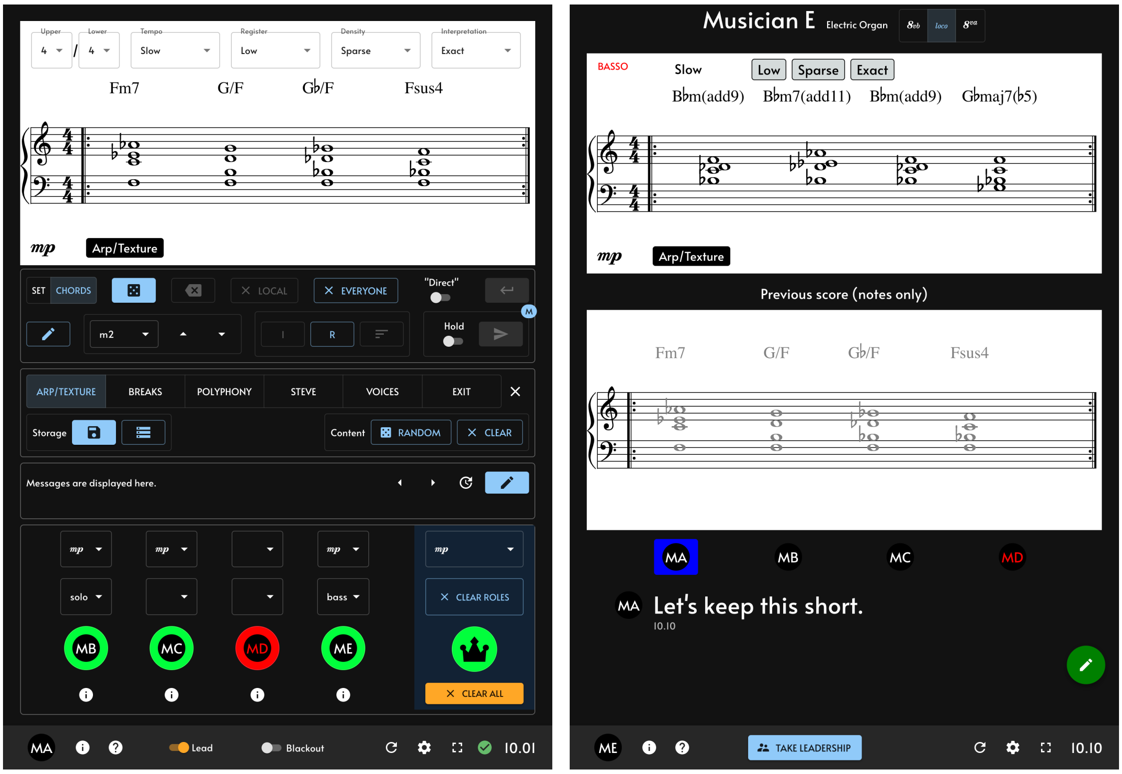

Figure 1. Two interfaces on a tablet: leader (left) and musician (right).

send a signal to indicate abandoning leadership, meaning that any another musician could jump to that role. However, in the current implementation the leadership can be taken by anyone at any moment, even during the blackout mode.

The screen is divided into five main sections. Below the score are various editing functions such as transposition by a selected interval, inversion and retrograde. With the switch labeled *direct* turned on, the score is sent automatically after a user-configurable period of inactivity, allowing to enter the score with a MIDI input device and distributing it for the musicians without touching the interface.

Below the editing functions are the keywords and buttons for the storage and content operations. A narrower section is reserved for text messages. Due to differences in available screen space, the text messages are displayed differently in the leader and musician interfaces. In the leader interface, a dialog window pops up when a new message arrives, requiring the leader to close the window before continuing with other tasks. In the musician interface, the messages are displayed in an auto-scrolling area similar to instant messaging applications, requiring no interaction from the musician.

The "mixer" at the bottom of the screen takes up most of the screen space, as it includes individual controls for all musicians that are displayed as avatar images or as initials if there is no image available. In its current design, the mixer can fit up to seven musicians (the leader is not displayed). The mixer area is used to set dynamics and roles for the musicians. Musicians can be also selected/deselected in order to send score or other parameters only to a selection of musicians. The master channel allows to change all dynamics or clear the roles for the selected musicians, as well as clearing all content with a single button.

2.2 Musician interface

The musician interface displays the score and other instructions received from the current leader, as well as text messages from any member of the group. Following the musician interface does not require any physical interaction with the device.

The header contains the musician's name and the instrument. For multi-instrumentalists a drop-down menu appears where it is possible to switch to another instrument with a correctly transposed part. A button allows to display the score either as transposed or in concert pitch, and an additional octave-up/-down transposition button is available to make it easier to read notation with many ledger lines.

The active score—the one to be played—is displayed below the header with the previous score placed below. A variation of a segmented score was selected (for an overview of various options, see [17, pp. 15–18]) for tablet use; on a smaller mobile phone screen there is enough space only for the active score, at least with instruments notated on a grand staff. It may feel counterintuitive to have the previous score displayed below the active one, but this arrangement was chosen to always have the active score in the same position in the leader and musician interfaces, and therefore provide continuity.

Similar to the leader interface, below the score are avatars for other musicians. The avatars can be clicked to see additional information about the musicians. The leader musician is framed with special colors which indicate if the leadership is active or inactive (“abandoned”). Finally, the lowest screen area is reserved for displaying text messages or entering the text message dialog window.

3. DESIGN PRINCIPLES

There are several principles that have affected the design of the application, most notably usability (learnability and efficiency) and cost-effectiveness. From the common definitions of usability (see for example [18, pp. 19–22]), learnability has been most the important design attribute for both the application and the whole concept. In this project, the importance of the learnability stems from the practicalities of working with professional musicians: they may not be able or willing to spend an inordinate amount of time learning complex applications, concepts and rules—or the signs and gestures of non-digital practices. Making the application easy to learn and use allows for rehearsal time to be spent more efficiently on making music with the application instead of learning to use it. Because the application can be used with a very limited initial configuration, learning can happen cumulatively by adding new parameters to the configuration as needed. For example, sending pitch sets or chord progressions entered via a MIDI input device could be the first step in the learning process. Alternatively and even more simply, dynamics could be the only parameter to be changed. After learning one or two features, more parameters could be incorporated based on the needs of the musicians (and the constraints of the application).

Efficiency is another usability attribute that has been taken into account in the design. First, the application is primarily designed for tablets, eliminating the need to move a pointer before clicking a user interface (UI) element. Second, within the limits of available screen space the UI elements have been optimized so that a minimum amount of interactions is necessary. This results in a somewhat crowded UI but affords a more efficient use of the application since most common operations can be launched directly instead of opening menus or dialogs.

Outside the realms of usability, cost-effectiveness has been another important factor, both for the development and for the user. The application uses only open source libraries and the development does not require any commercial software. For the user, the application works in a standard web browser (see, however Chapter 5). It also works on lower tier devices, although a more high-end device with a faster response does provide a better user experience.

4. DEVELOPMENT PROCESS

The aim of the development process has been to create a high-fidelity prototype [18, pp. 428–433] where the application would eventually evolve to a final product through continuous development, testing and evaluation. After the initial development stage based on the original ideas was largely complete, the application was taken to the first rehearsal session in September 2020. Since then, it has been tested and developed with feedback from the participating professional jazz musicians who are playing keyboards, guitars and drums. At this stage of the project, the instrumentation of mainly polyphonic instruments was selected so that all musicians except the drummer could take different roles such as bass, solo or chordal accompaniment.

All rehearsals produced log files and audio recordings that included both the played music and discussions in-between, providing immediate feedback about the system. For example, shortcomings of the application were collectively discussed right after they were found, as well as various strategies to deal with the information communicated with the application. Additional analysis was done after the sessions, for example, by scanning the log files for chords where the application had failed to generate a proper chord symbol, and fixing them for the next session. In addition, the music on the recordings was listened to with subjective aesthetic criteria: does the music work? If it does not, where is the problem?

Some development decisions were made by observing the musicians using the application. While fixing bugs is an ever-present feature of software development, optimizations were implemented after noticing that certain operations were requiring too many interactions. For example, editing the score required pressing an edit button before the editing functions were enabled. Keeping the editing functions always enabled eliminated the need for the edit button and subsequently made the application more efficient. Some features were implemented based on the suggestions made by the musicians, such as the score storage and the selection of roles (e.g. solo or bass), first quickly implemented through the use of keywords, but later refactored into a separate functionality.

5. TECHNICAL IMPLEMENTATION AND LIMITATIONS

Frankie consists of a web application and a server. The web client written in TypeScript uses React [2] and MUI [3] for common user interface components such as buttons, drop-down menus and so on. Other core components include Recoil [4] for state management, VexFlow [5] for notation rendering and JZZ [6] to handle the MIDI input happening via Web MIDI API. The communication between the server and the clients employs Apollo GraphQL [7] libraries. Excluding few experimental features, the Node.js [8] server does not currently modify the score or other parameters, and works mainly as a transmitter of data between the client devices. As an additional tool there is a log reader that converts the JSON log files into various formats for further analysis: PDF for easier

[2] https://react.dev/
[3] https://mui.com/
[4] https://recoiljs.org/
[5] https://vexflow.com/
[6] https://jazz-soft.net/doc/JZZ/
[7] https://www.apollographql.com/
[8] https://nodejs.org/

reading, CSV for importing markers into a DAW such as Reaper and SRT for creating subtitles to videos.

Since the WebKit browser engine does not implement Web MIDI API, [9] browsers based on WebKit cannot be used for MIDI input. These browsers include Safari and browsers for iOS and iPadOS devices where all vendors are required to use WebKit. However, WebKit browsers can be used if MIDI input is not necessary, for example with musicians who are not going to send staff notation. As of this writing, Google Chrome on an Android device is the recommended combination for MIDI use.

Due to the development focus on the usability of the user interface, the current server implementation can only run a single session at a time. For a proper user and session management, the server code needs a full rewrite and the addition of a database system. As a related limitation, authentication or authorization are not implemented, yet, so the system can be run most safely on a network that is not connected to the internet. As a result of these limitations and the general work-in-progress state of the whole system, it is not yet publicly available, either as source code or as a ready-to-use web service.

6. CASE: *LABRA*

The project was first presented publicly in *Labra*, [10] a concert held at the Black Box of Musiikkitalo in Helsinki on September 28, 2022. In this concert, a group of five musicians—organ player, keyboardist, drummer, electric guitarist and the author on another electric guitar and bass—were playing based on the material created and communicated in real-time with the application. In addition to their personal instruments, the musicians were equipped with various models of Android tablets and small two- or three-octave MIDI keyboards connected to the tablets either via Bluetooth MIDI or a USB cable (see Figure 2). Although operating a MIDI keyboard can possibly add an extra level of difficulty for non-keyboardists, a separate MIDI input device allows playing and simultaneously operating the application, for example, by sustaining a note or a chord on a guitar with a left hand while using the MIDI keyboard with a right hand.

The concert was prepared with five two-hour rehearsals that were scheduled so that some development of the application could be done between the rehearsals. These development periods were constrained only to bug fixes and optimizations, to avoid spending limited rehearsal time on learning new features. To try out the application outside the rehearsals, the musicians were given access to the demo version of the application deployed on the Heroku platform. [11]

6.1 General remarks

The concert performance lasted for 58 minutes without any breaks. Based on the JSON log file, there were 13 changes in the leadership and a total of 49 notated scores created and played during the performance. A single chord or a chord progression was the most popular score type; scores of the pitch set variety appeared only three times. Based on the rehearsals and musicians' preferences, the dominance of the chord progression score was to be expected. Eight of these chord scores were loops of 2–3 chords and the remaining chord scores were either single chords or pedal points sustained for a longer period. It had been found in the rehearsals that single chords and chord loops of 2–4 chords were short enough to quickly enter into the application during playing and had provided useful launching points for improvisation and further elaboration.

The parameters of time signature, tempo, density, register and interpretation were available in the drop-down menus, with 4–5 options for the text-based parameters and multiple options for the time signature. All parameters could be left undefined which was the default option. The parameter set had stayed fairly constant since the initial 2020 rehearsals that were held with partially different musicians. The interpretation parameter with the options [12] 'Completely free', 'Rather free', 'Rather exact' and 'Exact' was inspired by Butch Morris's conduction gesture *develop* which indicates the degree of development for an idea [13, pp. 178–179]. During the concert, tempo changes were communicated 6 times, density changes 8 times, register changes 7 times and interpretation changes 3 times. The only parameter not communicated with the application was the time signature, although it had been used in the rehearsals for various time signatures such as 3/4, 4/4, 7/4, 6/8, 9/8 and 12/8.

The keywords took their shape during the rehearsal period with 'Arp/Texture', 'Breaks', 'Polyphony', 'Steve', 'Voices' and 'Exit' ending up being used in the concert configuration. The meaning of these six keywords were discussed in the rehearsals. For example, 'Breaks' was inspired by the stop-time cues that Miles Davis used with his 1970s groups [19]. Two musicians used these keywords and except for 'Exit'—indicating a suggestion to exit the current musical idea—all keywords were used in the performance.

Free-form text messages were used by all musicians, 21 times in total. Compared to the keywords with predefined musical meanings, the text messages were sometimes more abstract such as 'Dry wood' or 'Backward jazz'. However, most messages were used for less ambiguous requests such as 'A single chord only' or 'Superfast beat with brushes shortly'. Although texting takes more time than using the predefined keywords or other more streamlined features, it was considered a very useful feature, with one musician comparing it to being as powerful as any other feature of the application.

While the screens were turned off with the blackout mode seven times by four different musicians, the score storage found more limited use, with a single content set saved to and sent from the storage during the performance. To summarize, most features of the application were used in the concert.

[9] https://webkit.org/status/#specification-web-midi

[10] *Labra* is a Finnish equivalent for the word *lab*, a laboratory.

[11] On November 28, 2022, Heroku discontinued their free plans and the demo version of the application is not hosted on the platform anymore.

[12] All predefined textual parameters used in the concert were in Finnish. For this paper, these have been translated into English.

Figure 2. Various placements of tablets and MIDI keyboards in *Labra*. The tablet screens have been turned off before the concert performance to save battery.

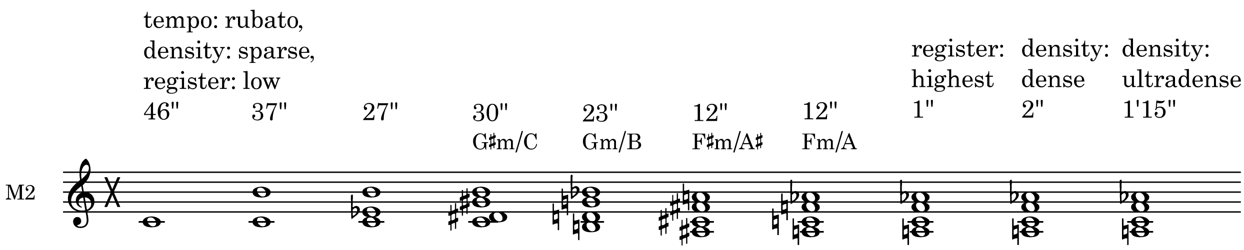

Figure 3. The first ten chords and parameters sent by musician 2 (M2) in *Labra*.

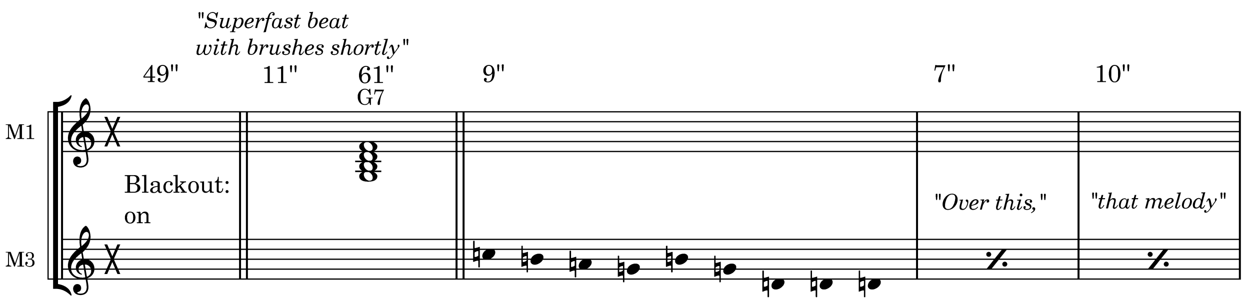

Figure 4. Changing leadership between musicians 1 (M1) and 3 (M3) in *Labra*.

6.2 Two examples

Figures 3 and 4 display two sections [13] from the log file as engraved with the Dorico notation software. The durations of the events are displayed above the staff in seconds. The first example (see Figure 3) contains the first ten chords and parameters sent with the application by musician 2 (M2) during the first ca. 4 minutes of the performance, after a short collective free improvisation had been played by the other musicians. The first event contains a single pitch (a middle C) and the parameters of tempo (*rubato*), density (*sparse*) and register (*low*). This slowly builds to a G#m/C chord which then begins to move downwards chromatically until reaching Fm/A. During that chord the parameters of register and density are changed again, resulting in *ultradense* Fm/A chord in the *highest* register of each instrument. This harmonic idea live-composed by M2 during the first minutes of the performance—a minor triad with a major third as the lowest note—resurfaced ca. 10 minutes later when musician 3 (M3) brought it back with the message 'A single chord only'. [14]

In the second example (see Figure 4) musician 3 (M3) has been leading the group for 4 minutes before abandoning leadership and turning on the blackout mode (the first event in the example). During the blackout mode musician 1 (M1) has taken the leadership and sent the message 'Superfast beat with brushes shortly' before giving the G7 chord as a harmonic guideline. After M1's short leadership, M3 retakes the control and sends a melodic fragment ('that melody') to be played over the current background ('over this'). While M1's leadership lasted for less than 90 seconds and contained only a single instruction for the drummer and a harmonic idea for the others, the music took a different direction both harmonically and rhythmically, and provided a platform for M3's subsequent melodic idea.

7. CONCLUSIONS AND FUTURE WORK

Based on the feedback collected from the participating musicians, [15] in its current state, the application has been relatively easy to learn and use, thus at least partly accomplishing one of the development goals. The application has enabled to shape music with established musical parameters without the need to have extensive training. Musicians have found it fun to make music with the method.

Since the method introduces a new modality for the musicians, this requires dividing attention between playing, listening and operating or following the application, a phenomenon recognized in previous research as well [7]. Musicians found it challenging to combine these modalities, although simply playing more gigs—not only rehearsals—was offered as a possible solution to becoming more fluent with the concept. To avoid becoming completely overwhelmed with the additional information it was found useful to abstain from playing while operating the application (see Chapter 6.2). Similar best practices still need to be found through more public performances.

Musicians also noticed a content gap between the musical ideas they communicated with the application and the realization that followed. Although the music did not always sound as they had imagined, this was not necessarily considered to be a negative thing. For example, one of the musicians explained having a "let's see what happens" type of attitude towards the material while another one reflected that a vague and unfinished idea may turn out to be the best moment of a concert, and that a better outcome would not be guaranteed by more "masterful" information.

Some audience members were missing the ability to follow the interactions, for example through the projection of the score or the application interface, a common practice in live coding performances [8]. The reasons for the decision to not offer score visualization for the audience are not unique for this project [8, 5] and are too manifold to be exhaustively described within this paper. However, when asked about the lack of projection, one of the musicians stated that it would have impacted the way the musicians approached the music making, for example by generating unwanted reactions in the audience. Nevertheless, the performative aspect still needs further consideration to keep audiences less puzzled.

As the application is still a work-in-progress, the future work includes typical programming tasks such as bug fixes, optimization and refactoring. The user interface is under constant revision to provide a better and more inspiring user experience, as well as responsive designs for different screen sizes and orientations. There are areas such as enharmonic spelling which are still suboptimal and in need of further development. As mentioned earlier (see Chapter 5), the server is being rewritten to allow easier access and more spontaneous use of the application. One of the most requested features by the musicians is the addition of durations, especially to the pitch sets. Designing an efficient method for live use is one of the next steps in the development of the application. Additionally, there have been a couple of experiments with server-side processors that modify the content, such as distorting the input pitch sets to give cluster-like results. More work will be done in this area later in the project.

Acknowledgments

The beginning of the project in 2020 was funded by a 12-month working grant from Kone Foundation. Without this grant the development would have not been possible. Later, University of the Arts Helsinki provided funding to organize the concert. Finally, I would like to thank Frankie the cat for standing in my sight when I needed a title for the application.

8. REFERENCES

[1] L. D. B. Morris, "Testament: A Conduction Collection," New York, NY, 1995.

[2] B. Faria, "Exercising musicianship anew through soundpainting: Speaking music through sound ges-

[13] The audio clips for Figures 3 and 4 can be listened to at `https://www.researchcatalogue.net/view/2008271/2008272`

[14] The reentry of the idea is not included in the notated example.

[15] The majority of the musicians were individually interviewed in semi-structured interviews which were transcribed, and analyzed especially on the themes of the method, authorship and the usability of the application.

tures," Ph.D. dissertation, Lund University, Lund, 2016.

[3] J. Zorn, "The Game Pieces," in *Audio Culture: Readings in Modern Music*, C. Cox and D. Warner, Eds. New York, NY: Continuum International Publishing Group, 2005.

[4] J. Brackett, "Some Notes on John Zorn's Cobra," *American Music*, vol. 28, no. 1, pp. 44–75, 2010.

[5] V. Goudard, "John, The Semi-Conductor: A Tool for Comprovisation," in *Proceedings of the 4th International Conference on Technologies for Music Notation and Representation*, S. Bhagwati and J. Bresson, Eds., Montréal, 2018, pp. 43–49.

[6] S. Kalonaris, "Adaptive Specialisation and Music Games on Networks," in *Proc. of the 13th International Symposium on CMMR, Matosinhos, Portugal, Sept. 25-28, 2017*, Matosinhos, 2017.

[7] P. Dahlstedt, P. A. Nilsson, and G. Robair, "The Bucket System - a computer mediated signaling system for group improvisation," in *Proceedings of the International Conference on New Interfaces for Musical Expression, Baton Rouge, LA, USA, May 31-June 3, 2015*, Baton Rouge, 2015.

[8] S. Wilson, N. Lorway, R. Coull, K. Vasilakos, and T. Moyers, "Free as in BEER: Some Explorations into Structured Improvisation Using Networked Live-Coding Systems," *Computer Music Journal*, vol. 38, no. 1, pp. 54–64, 2014.

[9] S. Zagorac and M. Zbyszynski, "Networked Comprovisation Strategies with ZScore," in *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2020*, R. Gottfried, G. Hajdu, J. Sello, A. Anatrini, and J. MacCallum, Eds. Hamburg: Hamburg University for Music and Theater, 2020, pp. 133–140.

[10] D. Andersen, "Indra: A Virtual Score Platform for Networked Musical Performance," in *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2020*, R. Gottfried, G. Hajdu, J. Sello, A. Anatrini, and J. MacCallum, Eds. Hamburg: Hamburg University for Music and Theater, 2020, pp. 227–233.

[11] J. Burrows and C. G. Reed, "Free Improvisation as a Path-dependent Process," in *The Oxford Handbook of Critical Improvisation Studies, Volume 1*, G. E. Lewis and B. Piekut, Eds. New York: Oxford University Press, 2016.

[12] J. P. Davis and K. M. Eisenhardt, "Rotating Leadership and Collaborative Innovation: Recombination Processes in Symbiotic Relationships," *Administrative Science Quarterly*, vol. 56, no. 2, pp. 159–201, 2011.

[13] T. T. Stanley, "Butch Morris and the Art of Conduction," Ph.D. dissertation, University of Maryland, 2009.

[14] S. Bhagwati, "Comprovisation – Concepts and Techniques," in *(Re)Thinking Improvisation: Artistic Explorations and Conceptual Writing*, H. Frisk and S. Östersjö, Eds. Lund, Sweden: Lund University, 2013, pp. 99–104.

[15] A. C. Beal, *Carla Bley*, ser. American Composers. Urbana: University of Illinois Press, Nov. 2011.

[16] S. Hayden and M. Kanno, "NEXUS: Live Notation as a Hybrid Composition and Performance Tool," in *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2020*, R. Gottfried, G. Hajdu, J. Sello, A. Anatrini, and J. MacCallum, Eds. Hamburg: Hamburg University for Music and Theater, 2020, pp. 13–23.

[17] S. Shafer, "Recent Approaches to Real Time Notation," Ph.D. dissertation, University of North Texas, Denton, May 2017.

[18] H. Sharp, Y. Rogers, and J. Preece, *Interaction Design: Beyond Human-Computer Interaction*, 5th ed. Indianapolis, IN: John Wiley and Sons, 2019.

[19] D. Liebman, "Miles and Me," https://davidliebman.com/home/ed_articles/miles-and-me/, 2009.