Composing Interface Connections for a Networked Touchscreen Ensemble

Charles Patrick Martin
Australian National University
Canberra, Australia
charles.martin@anu.edu.au

Alexander Hunter
Australian National University
Canberra, Australia
alexander.hunter@anu.edu.au

Brent Schuetze
Australian National University
Canberra, Australia
brent.schuetze@anu.edu.au

Yichen Wang
Australian National University
Canberra, Australia
yichen.wang@anu.edu.au

***Abstract*—In this paper, we explore new ways to compose networked interactions for ensemble performance using touchscreen instruments. We introduce a web-based touchscreen instrument modelled on classic synthesiser designs and a system for linking interface elements from this instrument between multiple members of the ensemble. We discuss our composition system and syntax that allows these network links to be scripted allowing a mode of performance inspired by open-form composition. The experience of composing and performing with these systems is investigated through co-located live and studio performances. Our findings support the utility of the system for improvisations and suggest ways to improve the system to cope with larger numbers of network links to enable more exciting performances.**

***Index Terms*—touchscreen, networked music, collaboration, improvisation, open-form music**

I. Introduction

Networks are often used to keep time between electronic instruments but other ways of synchronising performances have not been explored to the same extent. Networked aspects of performance are generally hard-coded into instrument designs, rather than composed in collaboration with musicians. The goal of this project is to investigate a compositional language for open-form music with networked musical instruments that enables flexible exploration of network synchronisation of interface elements. We have built a software system for coordinating open-ended musical performances between multiple co-located performers creating music on touchscreen computers. This system allows connections between these instruments to be formally specified and automatically scheduled during open-ended performances. Further, our system allows the composed network connections to be edited and updated from a web-based user interface in between performances so that a composer can interact live with an ensemble. In this paper, we introduce these software systems and discuss how they can be used to create open-form performances. We discuss a case study of live and studio performances where we experimented with the formal language for specifying network connections during performance. Our findings suggest that the density of connections had the most impact on the ensemble performance and that randomness in specifying rules can work well in our improvisations.

This research is supported by the Commonwealth of Australia as represented by the Defence Science and Technology Group of the Department of Defence.

In the context of this work, we frame a networked ensemble as a group of performers where aspects of the electronic musical instruments are connected via a network. In a simple case, this means that when a musician changes a parameter (e.g., the volume of an oscillator), this change is communicated across the network and effected on other performers' instruments. We call such a link an interface connection. More complex cases are possible, for instance where different UI elements are connected, where one-to-many connections are made, or where parameter changes are inverted or delayed in time. We aim to use this system to encode or enhance collaborative concepts among the ensemble such as coordination, opposition, synchrony, bandwidth limitations and delay.

These interface connections can be dynamic during a performance; new connections can be established, modified or removed. To frame our work musically, we are adapting the concept of open music to this networked ensemble context. In open music, works are not complete until performed, that is, some aspects of performance are composed and others are decided in the moment by musicians. Our open compositions define a schedule for interface connections to be established and removed while other aspects of the performance are freely improvised. In this way, the network connections form a changing environment for improvising performers to work within, with challenges or affordances presented by these connections over time. We aim to explore how composing with these connections can affect the "group mind" experience of improvising in an ensemble [1].

Our system has three components shown in Figure 1:

1) *Synth Interface:* A web-based musical instrument that can be linked by the server. The interface accepts input from a performer through a touchscreen and outputs sound directly through the device's internal speakers or an audio system. The configuration of the interface can be changed remotely.
2) *Composition Interface:* A web-app for entering and controlling an open-form composition that coordinates changes to interface connections over time. The composition client allows musical actions of performers to be influenced by an open-form score represented as YAML code.

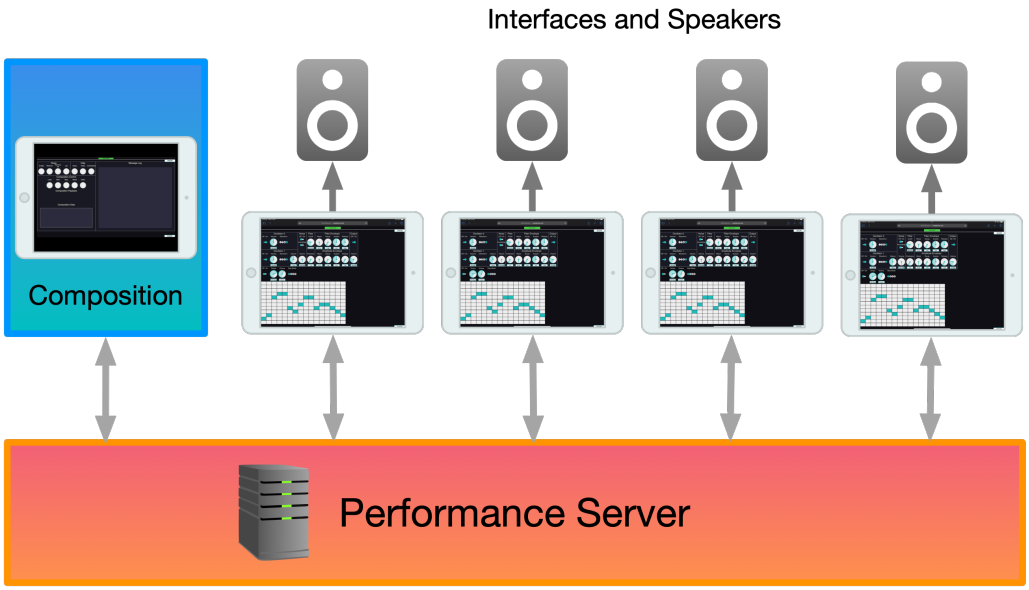

Fig. 1. An overview of our system design. Four iPad computers run the music interface. The composition client schedules changes to interface connections between the performers and the server coordinates data transfer between all clients. Our system is available online at http://metatone.net

3) *Server:* A web-server that can connect to interface and composition clients, and coordinate the exchange of information between them. The web-server is designed to be general so that it can be used in a variety of networked music contexts.

In the remainder of this paper we articulate our concept of open compositions for a networked touchscreen ensemble and discuss the relationships with other research in networked music. We discuss our system design and pay particular attention to the compositional syntax that defines interface connections in time. Our system has so far been evaluated through a prototype live performance and two studio rehearsal sessions performed by the authors. These experiences demonstrate that our web-based musical instrument and server designs are feasible for open-form networked ensemble performance. Further, our work shows that such performances are driven by the density of interface connections more so than specific connection configurations. This result suggests that a composition syntax focussed on connection density may be more appropriate for future exploration.

II. Background

This research is focussed on networks of co-located electronic music performers. Networked ensembles have existed since the arrival of portable, personal computer systems [8], [21]. Powerful mobile computers such as smartphones enabled larger ensembles, more rapid development, and experiments in performance through the affordances of touchscreens [15]. The COVID-19 pandemic has reignited the existing development of remote networked ensembles creating collaborative works with software systems such as L2Ork Tweeter [2], DIY hardware controllers [19], democratised music technologies such as webcams [4], live coding systems [16], VR environments [3], and mobile web apps [20].

Some of these examples provide time synchronisation for performing pulse-based musical idioms and commercial solutions such as Ableton Link are used widely. Other aspects of music can also be synchronised during a musical performance, for instance, the scale and sounds available in a touchscreen music interface [11], [12]. Our system similarly allows synchronisation of non-rhythmic elements of a musical instrument, but we extend this idea to connecting any interface element in a web-based synthesiser and in terms of treating these connections as a compositional material.

The musical intention of our work is to explore semi-structured improvisation, taking inspiration from the open-form works of American composer Earle Brown [6]. In open works, some aspects of the composition are left to the performers to decide. While Brown used written scores to communicate instructions, networks of musical devices have also been used to similar effect. Dahlstedt et al.'s Bucket System [5] allows improvisation instructions to ensemble members to be suggested and displayed on a MIDI interface. Hope and Vickery's Decibel Score Player [9] allows any graphical or notated score to be simultaneously scrolled across the screens of members of an ensemble. In this research, we also allow for a networked open-form composition. In contrast to some previous examples, our composition is implicitly represented through interface connections in our app and not displayed as instructions to performers.

III. System Design

This section describes the design of each part of our open-form music performance system. Our project is open source

Fig. 2. The synthesiser interface running on an iPad touchscreen computer. This includes a representation of a typical synthesiser with a simple step sequencer for creating a loop of notes. The synth interface is available online at http://metatone.net/model-d/

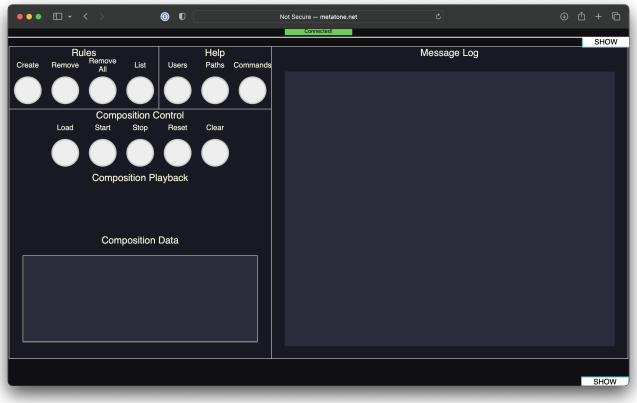

Fig. 3. The composition interface including a text field for a composition to be entered. The buttons allow limited manual control over the server (for instance, adding random rules) and transport controls for the composition timer. The composition interface is available at: http://metatone.net/control/

and the code is available online[1]. The system can be used online at http://metatone.net or locally hosted using Docker.

A. Synth Interface

The computer music interface in our system is a web-based musical instrument designed to run on a touchscreen computer such as Apple's iPad Pro. The idea behind the interface is to emulate a typical synthesiser and sequencer setup for basic loop-based musical production. Sound output in our system is diffused either from built-in speakers on the touchscreen computers or through a monitor speaker connected to an audio interface.

The synthesiser section reconstructs a classic two-oscillator subtractive synthesiser. The control layout is similar to a Minimoog Model D (or this synth's many clones). The sequencer allows the creation of a looped, monophonic melody in a grid interface similar to a TenoriOn [14] and other grid-based instruments such as the Monome, Launchpad or ZOIA [17]. The sequencer has parameters for tempo, octave and step-mode where the options are forward, backward, random walk and random. The lack of a piano-style keyboard was an intentional design choice to encourage performers to focus on parameter tweaking rather than playing individual notes.

The interface is constructed in HTML/JavaScript and makes use of the NexusUI [18] JavaScript library for displaying standard electronic music user interface widgets (such as knobs, switches and faders). For sound creation, the `tone.js` [10] library provides a web audio framework for defining synthesisers and sound sequencing. The interface communicates with the server via websockets.

The idea of the interface that each individual performer has autonomy in terms of defining and creating sounds, but is limited to one "instrument" or setup, and cannot create multiple melodies simultaneously. That is, they can only play lead *or* bass *or* keys, not all parts at the same time and thus must interact in an ensemble to create a complex and varied performance. All of each performer's interactions with their interface are sent to the server which governs how messages are sent to other players in the group.

B. Server

Our performance server is a back-end web application built in Python that is designed to interchange messages between an arbitrary number of clients during a musical performance. The server communicates with clients using the WebSocket protocol, this allows bidirectional communication throughout a performance and is more robust than other networking technologies frequently used in music technology (e.g., UDP sockets and zero-configuration networking) to the limitations of an enterprise networking environment. The server allows any interface web app to connect to a current performance. The server's configuration involves creating rules that define an interface connection. Each rule can link any parameter to any other between any two performers' interfaces. The server can further delay, delete or invert messages. The server expects to receive messages in a format inspired by Open Sound Control (OSC) [7].

C. Composition Interface

The composition interface (shown in Figure 3) is another web application, separate to the performance interface. This application allows the server to be controlled remotely and for a composition of interface connections to be notated in YAML text format. A composition is defined to be a list of events with a starting time and duration (both measured in seconds). Each event defines a list of interface connections with source and destination performers and interface elements (written in OSC-like address format). Each connection can optionally be

[1]Code available at: https://gitlab.anu.edu.au/u4110680/open-form-music-computing

inverted or subject to a delay. A simple composition containing one event with one connection could be as follows:

```
composition:
  -
    time: "00:00"
    duration: "00:20"
    connections:
      -
        source: player-a
        source-path: /synth/env/attack
        dest: player-b
        dest-path: /synth/env/sustain
        invert: false
        delay: 0.5
```

The source and destination player and path can both be set to be random in which case the server connects two random interface elements from random performers.

In addition to the composition text field, the interface includes buttons to start and stop compositions and for adding and removing rules. A console shows composition activity and performers joining the server.

The idea of having a composition "interface" rather than just hard coding a composition into the server or synthesiser apps is to allow for exploration of networked ensemble open-form composition actively during performances and rehearsals. This system allows a composer to define a composition ahead of time, edit it in between rehearsals (without access to server or application code) and to interact in real time during performance.

Fig. 4. Studio performances with the prototype system. From top to bottom: studio duo improvisation session to develop composition interface, studio quartet to test composition system. Video of the quartet session is available online at https://youtu.be/oAfGLf5dwkU

IV. Practice and Performances

The process for exploring our system has so far been autobiographical [13] where the system has been explored primarily by the authors and used in our performances and recordings (see Figure 4). In this section we discuss the experience of a live concert and two recording sessions. The concert involved prototype versions of the interface and network server and did not involve the composition system. In this case, network connections were live-coded by interacting with the server application using Python commands. The performance was framed as a proof-of-concept and demonstrated that the synthesis interface and server were working. The live-coding modality enabled network connections to be created throughout the performance; however, we noticed that it was difficult to generate enough connections manually to make a strong impact.

We addressed these limitation in our studio improvisation sessions where we used the composition interface to schedule networked interface connections during the performance. This process involved two recording sessions with two and four performers respectively. The first session focussed on exploring the capabilities of our composition syntax and creating an initial composition for exploration by a group of four. In this session we realised that random any-to-any rules, where any UI elements from any two performers can be linked were most useful in planning a composition for a flexible group.

The second session aimed to find out how well this composition worked in a collaborative improvisation. Similarly to the earlier concert, we noticed that many more networked interface connections than expected were needed to make an impact on improvisation. It appears that the density of interface connections is more important than specific connections in this open-form idiom where the style of performing from any one performer is not known in advance. This empirical realisation is supported by the theoretical mapping problem: more performers dramatically increases the number of potential interface connections. If a single UI connection is established, it is unlikely that this will happen to be between elements that are the current focus of any two performers. So a large number of connections are needed in order for these to be discovered and used by performers in the improvisation.

During the session we worked around this issue by replicating rules in our composition until they were definitely noticeable in the performance. Having done so, we noticed that the UI connections could be discovered and actively used by performers to adjust not only their own sound, but the sounds of others. The trade off was an unwieldy composition file of 8000 lines. This suggests that our syntax for compositions

needs to adapt to allow rules to be replicated within one line. Perhaps the rule density could be a more interesting compositional variable than the ability to link specific performers.

The rehearsal sessions also verified that the synthesiser interface was capable of a wide range of sounds and that our simple sequencer design allowed expression within the open-form music idiom. Further tweaking of the interface ranges and scales for the synth could allow new users to have a more productive improvisation as we spent early sessions learning and discovering expressive performance configurations.

V. Conclusion

This work has introduced a system of performance interface, server, and composition interface for creating open-form performances for ensembles of touchscreen performers. Our system defines a new concept for composing using interface connections as a primary way of guiding an improvising ensemble. We described the details of our system and the syntax for defining compositions and reported on our initial autobiographical artistic research experiments. The findings of our work supports the utility of our synthesiser interface in open-form performances and verified that our server and composition system can be used to interact with and influence performers. We found that randomised rules were most practical in creating these open-form compositions but that a more compact syntax for defining large numbers of rules is required.

Our future work for this project includes improvements to the synth interface and composition system ahead of a performance study with a wider pool of performers. We plan to adapt the synth interface for an improved improvisation experience, particularly to allow new users to get started quickly. The interface could make it clear when networked connections are in place, e.g., by highlighting UI elements. The composition language may need to be rethought to focus on rule density so that editing can take place efficiently in between performances. Our composition system could also allow rules to be tweaked in real-time either by a composer agent or a human. Future studies with this system will focus on isolating the effects of different compositional techniques on the performers and their music. For instance, delayed or distorted messages may have different effects than straightforward mappings.

References

[1] D. Borgo. *Sync or Swarm: Improvising Music in a Complex Age*. Continuum, New York, NY, USA, 2005.

[2] I. Bukvic. Latency-, sync-, and bandwidth-agnostic tightly-timed telematic and crowdsourced musicking made possible using l2ork tweeter. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, The University of Auckland, New Zealand, jun 2022. doi:10.21428/92fbeb44.a0a8d914.

[3] C. Cakmak, A. Camci, and A. Forbes. Networked virtual environments as collaborative music spaces. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 106–111, Brisbane, Australia, 2016. Queensland Conservatorium Griffith University. doi:10.5281/zenodo.1176002.

[4] D. Cocchiara and L. Turchet. Democratizing access to collaborative music making over the network using air instruments. In *Proceedings of the 17th International Audio Mostly Conference*, AM '22, page 211–218, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3561212.3561227.

[5] P. Dahlstedt, P. A. Nilsson, and G. Robair. The bucket system — a computer mediated signalling system for group improvisation. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 317–318, Baton Rouge, Louisiana, USA, May 2015. Louisiana State University. doi:10.5281/zenodo.1179048.

[6] E. Dubinets. Between mobility and stability: Earle Brown's compositional process. *Contemporary Music Review*, 26(3-4):409–426, 2007. doi:10.1080/07494460701414405.

[7] A. Freed and A. Schmeder. Features and future of open sound control version 1.1 for nime. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 116–120, Pittsburgh, PA, United States, 2009. doi:10.5281/zenodo.1177517.

[8] S. Gresham-Lancaster. The aesthetics and history of the Hub: The effects of changing technology on network computer music. *Leonardo Music Journal*, 8:39–44, 1998. doi:10.2307/1513398.

[9] C. Hope and L. Vickery. The Decibel ScorePlayer — a digital tool for reading graphic notation. In *International Conference on Technologies for Music Notation and Representation*, TENOR 2015, Paris, France, May 2015. IRCAM. URL: http://tenor2015.tenor-conference.org/papers/08-Hope-Decibel.pdf.

[10] Y. Mann. Interactive music with tone.js. In S. Goldszmidt, N. Schnell, V. Saiz, and B. Matuszewski, editors, *Proceedings of the International Web Audio Conference*, WAC '15, Paris, France, January 2015. IRCAM.

[11] C. Martin and H. Gardner. That syncing feeling: Networked strategies for enabling ensemble creativity in iPad musicians. In *Proceedings of CreateWorld*, Brisbane, Australia, Feb. 2015. Griffith University. URL: http://hdl.handle.net/1885/95216.

[12] C. Martin, H. Gardner, and B. Swift. Exploring percussive gesture on iPads with Ensemble Metatone. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1025–1028, New York, NY, USA, 2014. ACM. doi:10.1145/2556288.2557226.

[13] C. Neustaedter and P. Sengers. Autobiographical design in HCI research: Designing and learning through use-it-yourself. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 514–523, New York, NY, USA, 2012. ACM. doi:10.1145/2317956.2318034.

[14] Y. Nishibori and T. Iwai. TENORI-ON. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 172–175, Paris, France, 2006. doi:10.5281/zenodo.1176979.

[15] J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang. Evolving the mobile phone orchestra. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 82–87, Sydney, Australia, 2010. doi:10.5281/zenodo.1177871.

[16] C. Roberts, K. Yerkes, D. Bazo, M. Wright, and J. Kuchera-Morin. Sharing time and code in a browser-based live coding environment. In *Proceedings of the First International Conference on Live Coding*, pages 179–185, 2015.

[17] B. Rossmy and A. Wiethoff. Musical grid interfaces: Past, present, and future directions. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Shanghai, China, June 2021. doi:10.21428/92fbeb44.6a2451e6.

[18] B. Taylor, J. Allison, W. Conlin, Y. Oh, and D. Holmes. Simplified expressive mobile development with nexusui, nexusup, and nexusdrop. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 257–262, London, United Kingdom, June 2014. Goldsmiths, University of London. doi:10.5281/zenodo.1178951.

[19] S. Thorn. Telematic wearable music: Remote ensembles and inclusive embodied education. In *Proceedings of the 16th International Audio Mostly Conference*, AM '21, page 188–195, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3478384.3478386.

[20] L. Turchet and M. De Cet. A web-based distributed system for integrating mobile music in choral performance. *Personal and Ubiquitous Computing*, 2023. doi:10.1007/s00779-023-01709-0.

[21] G. Weinberg. Interconnected musical networks: Toward a theoretical framework. *Computer Music Journal*, 29(2):23–39, 2005. doi:10.1162/0148926054094350.