DistributeDJ: a mobile group music making toolkit

Daniel Shapira, Shlomo Dubnov
Department of Music, University of California San Diego

***Abstract*—The activity of technology enhanced group music making can offer intriguing insights into musical interaction, new ideas on performance, and more easily provide non expert musicians the opportunity to participate. Collaborative improvisations often have a looser structure then traditionally composed score based music, but in most cases there are still sets of structural frameworks to be followed. DistributeDJ, a group music service, is proposed to aid in the creation and execution of flexible collaborative mobile music performances and attract a broader audience participation. Designed to run through an Internet client, the system provides a communication layer between a central performance server and multiple clients that are used to actively manipulate the contents of delivered audio through participant's mobile interface.**

I. The Role of Network Technology in Collaborative Music

Most of the online music services today provide the cloud equivalent of storage, playback and search functionalities that a regular desktop would normally have. These are mostly personalized applications that can benefit from additional information provided by other users and access to semantically annotated music data. As more and more content is put on the web, and as web and traditional entertainment delivery channels converge (such as web and TV), more and more scenarios of computing take the role of a remote control on an externally deployed content, with various degrees of interactivity and possibility for socializing or group interaction. Mostly known in online multiplayer games, the design elements of such applications require a centralized model with continuous persistence and automated execution of the event logic. The server structure usually consists of communication, simulation logic and object store layers that are responsible for executing the program code while providing the users with the elements they can manipulate while routing the data between the clients, the server and the application logic.

There are numerous benefits of computer mediated communication systems in the text based social and business world, and many of their concepts are applicable when making collaborative group music. The function of these group interactions is to modify a shared object. In the case of group music making, the shared object is the composition and each participant is altering part of the output from that composition. These emerging musical content models, sometimes known as improvisation sets, provide the user with a collection of musical elements that can be altered in real time to produce various musical results. Ranging from simple crossfading in a DJ setting, to track remixes and clip mashups, the manipulation of sound includes triggering of samples and continuous control of sound processing elements through a centralized sequencer application that mixes the sources in a way that maintains beat synchronicity under a common tempo and tonality. So far, the computer improvisation practices were limited to a single user or artist who manipulated the music through a digital audio workstation (DAW). In this paper we propose the architecture for an online deployment of a DAW that is used in a live playback scenario with control elements distributed across multiple users.

There are different models for communication in collective performance, each offering advantages and disadvantages over the other. Some provide more interaction potential and improvisational freedom, while others suggest a more organized performance with a firmer control structure[9]. The principle difference between them is the structure and directionality of the communication layer. A technological, computer based collaborative performance system can offer the capability to not only quickly modify its communication structure when needed, but also obfuscate the specifics of these structures from overwhelming non expert performers.

II. Existing group improvisation technologies

A. Collaboration solutions for mobile devices

Mobile music systems can be separated into two structural categories. The first group uses different programs in the production chain, starting with the mobile interface application, message routing and mapping tools on a receiving computer, and an audio production application such as a DAW or other music program. The second group is a unified solution with all interaction and sound production occurring on the mobile device. Both setups have advantages and disadvantages, with a most optimal choice different for each situation.

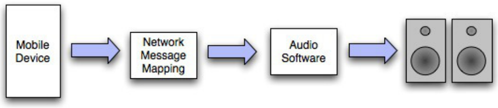


Figure 1. Elements of a multiple application production chain

1) Multiple application solutions: There are several network control applications available on the mobile marketplaces. TouchOSC [24] is the most mature of these with the largest feature set. There are a great many control elements available to an interface builder for creating custom files that are manually uploaded and placed in TouchOSC's mobile file structure. Very sleek and creative user experiences can be made with this application.

Once a user has manipulated an interface element and the data is received by the computer, special software is often needed to map the interface messages into a form understood by the audio programs. Osculator [16] functions well at this task, although its cumbersome setup time could be a roadblock for budding enthusiasts.

A Digital Audio Workstation (DAW) is primarily designed as a platform for recording, editing, and playing digital audio, but virtual effects and instruments have expanded their capabilities into realtime synthesis engines as well. If more custom audio processes are needed, interactive programming languages such as Max/MSP [14] or Pure Data [5] can be used.

The main advantages of the multiple application solution are higher quality tools for the user interface and audio production elements due to dedicated developers spending years refining the programs. Drawbacks of this group include increased setup overhead, more potential for system failure because of the multiple communication points, loss of portability, and some potential lack of customization.

2) Consolidated application solutions: Single application collaborative music solutions, those run within a single mobile device, have a different set of limitations and advantages. Once a mobile app is created, setting it up and running it is trivial in comparison to the multiple application solution. More flexibility in implementing unique interface ideas is also possible since the instrument is created using lower level programming languages instead of the interface templates of TouchOSC.

The consolidated solution is often at a disadvantage in the areas of performance coordination and audio fidelity. Without a centralized computer receiving the data packets from the networked performers, there is no opportunity to combine, filter, or otherwise organize the information streams in an informed manner. This situation resembles a more traditional music ensemble performing without the benefit of a conductor [25]. Audio fidelity can also be an issue when producing sound on a mobile device. The resolution and sampling rate does not currently match what is possible on a full computer, and reproducing audio using the device's built in speakers can be undesirable in most circumstances. There have been attempts by dedicated Mobile Phone Orchestras to create wearable sound amplification systems to improve this limitation [13].

Specialized mobile music programming frameworks have been developed to support the rapid creation of instruments for mobile phone orchestras. The MoMu Toolkit provides Objective-C based high level access to device sensors, networking, interface graphics, and real-time audio on the iOS platform [3]. urMus is a similar framework for iOS and Android platforms and with modifications accessible using C/C++ or higher level Lua scripts [6] .

III. DistributeDJ

A. A shift towards adaptable instruments and compositions

Computer Scientist Lynn Andrea Stein's 1999 article "Challenging the computational metaphor: Implications for how we think" described the common outlook within her field [20]:

> *Computation is a function from its inputs to its output. It is made up of a sequence of functional steps that produce- at its end- some result that is its goal.*

Common perspectives in music composition can be defined similarly, with the piece made up of rigid functional steps that are designed to produce a specific goal. The DistributDJ research is instead interested in changing ongoing behavior, redefining composition as necessarily interaction. Repurposing Stein's words to the musical context we seek to create situations where "the *composition* cannot be said to reside in any one of the entities; instead, it is the result of the interactions among them."

B. DistributeDJ system overview

The current structural setup of DistributDJ is similar to other multi application solutions, providing both a client interface and message mapping capabilities [22]. The main research goals of the project are to promote accessibility for wider adoption of mobile collaborative performance and increased real-time interaction through an information feedback channel that actively manipulates the contents of participant's instrument interface.

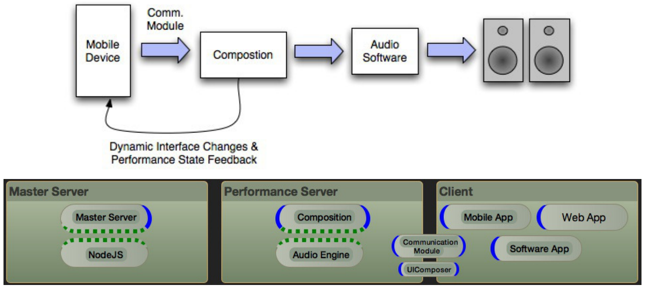


Figure 2. Overview of DistributeDJ's components with communication channels highlighted

C. DistributeDJ Components

1) Master Server: The Master Server functions as the dedicated initial communication point for a user searching for a collaborative performance to join. Upon running the DistributeDJ Client application, the Master Server is queried and provides the user with a listing of currently running group performances. The user simply selects their desired performance and the Master Server connects the Client to the selected Performance Server.

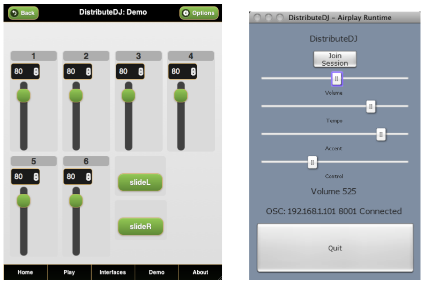
2) Audio Engine: The Audio Engine is what produces the audio result of the group collaboration and amplifies it back to the performers and audience. It's input parameters are controlled by the Composition via the Communication Module.

The specifics of an audio engine and it's input parameters are flexible and can be left up to the individual composer to construct. The Composition will have a set of tools available to aid in rapid construction of an audio engine using popular software programs such as Ableton Live and Max/MSP.

3) The Composition: The Composition is the information core of the group performance. It's main functionality includes:

- Setting up and dynamically pushing the user interfaces to the Client apps.
- Playing through a pre-composed DistributeDJ score. A score can contain definitions for sample playback, logic based parameter mappings, new UI configurations, messages to individual performers, and many more commands.
- Relaying mapped user input to the Audio Engine's input parameters.

4) Client Applications: The Client is the interaction point of contact for the user while they participate in the group performance. The DistributeDJ system is setup to increase the ease of this collaboration. The mobile applications are built using the Marmalade cross-platform SDK to increase portability to multiple device operating systems without the need for code base fragmentation. The web based client is built using a combination of the jQuery mobile framework on the front end, backbone.js for program structure, and the socket.io library for faster client/server network communication. Use of these cross-platform tools can lead to wider adoption/distribution and simpler application maintenance[2].



Volume 525

OSC: 192.168.1.101 8001 Connected

Figure 3. Software Client on left and Mobile Client on right

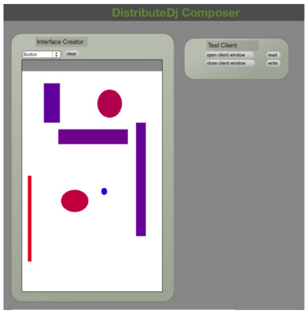


Figure 4. UI Composer

5) UI Composer: The UI Composer is an useful tool during the creation of a mobile group collaborative performance. When designing the score, multiple user interfaces can be rapidly created and stored in order to be later pushed onto the Client Apps at specified sections of the performance. This rapid interface development and flexible serving up of adaptable and individually addressable UIs allows for multitudes of new ideas to be discovered in a group setting as well as more established musical knowledge conveyed through hands-on exploration of the structures of music.

6) Communication Module: The Communication Module is the Performance Server's data conduit between the Composition, the multiple performing Clients, and the Audio Engine. The User Datagram Protocol (UDP) is currently used for sending and receiving data packets over the network due to its increased time-sensitivity over other protocols [21].

D. Further research

1) Social awareness mechanisms : The addition of a communication back-channel in the DistributeDJ system allows for many powerful socially communicative possibilities. This information feedback can be integrated into the visual display of input controls, such as a background color or border of a slider element. In addition to the implicit cues of the collaborative audio stream, a slider border that grows wider/darker when more users are actively interacting with that same element can be a very useful explicit social awareness cue. Other mechanisms such as a translucent indicator of an element's average group value may be desirable to inform participants about what is happening and what other performers are doing.

2) Automatic detection of conversational breakdowns: DistribteDJ's Composition Module holds the greatest amount of current information about the precise state of the system interactions. Metrics can be developed to detect potential breakdown in the ongoing musical conversation and steps can be taken to remedy the situation. For example, the value range of a interface element can be lessened to temporarily bring participants back to a narrower common context. While interfering with improvisational freedoms is not always desirable, having corrective measures in place can be beneficial and the choice left to the composer or ensemble.

3) Personalized interface adaptability: The flexible and dynamic changes to the user interface made possible by DistributeDJ can be improved further through personalized interface adaptability and real time gesture mapping. For example, if a performer has historically be interacting within a certain range on a slider, the system can adapt to give that performer increased resolution within that range. Real time gesture mappings have been proposed as a method of increasing the personalized control an improvisor has within a computer music system. A Hidden Markov Model solution is suggested by Wilson and Bobick [26], but much of the current research centers around real-time training of neural networks [4] [7].

4) Importing MusicXML into the Composition environment : A possible feature of the DistributeDJ Composition module would be the ability to import MusicXML or other musical score formats [15]. The loaded musical structures can then be manipulated or even edited in real-time by the performers.

5) Peer-to-peer performance capability: In addition to many user collaborative performances, simpler configurations of DistribteDJ can potentially be used in a smaller one-on-one setup without the use of a centralized performance server. These peer-to-peer situations will require the development of a number of complex client systems, but the effort may prove beneficial towards the project's goal of making collaborative music making more accessible.

IV. Uses outside the performance space

A. Music education

Xenakis saw the education potential of the UPIC and similar tools from its inception. Interactive technology such as interactive chalkboards are being used in schools with measurable success [1]. Targett and Fernstrom created a pair of musical games and tested their positive effectiveness that suggests prolonged use of such games could bring about the acquisition of various skills such as increased memory or the perception of pitch relationships [23]. The DristributeDJ and similar tools can be a viable method for bringing innovative music education interactivity to the classroom.

B. Music and sound exploration

Hope et al. suggest that interactive information systems in museum do not properly address social considerations, particularly when dealing with family dynamics [11]. Optional syncing of audio playback within the group, location based interactions, and group interest indicators are some features achievable with DistributeDJ or similar tools. Some related situations are sharing sound tags at specific locations within a city[12] or the using technology for a collaborative scavenger hunt [8].

C. Music Therapy

The simple act of being exposed to music and having limited involvement with its production has been shown to have health benefits such as stress reductions in heart patients [10] and can help with with communication skill of children with special needs. Clinical improvisation is a specialized form of music therapy improvisation and is defined by the Association of Professional Music Therapists as "musical improvisation with a specific therapeutic meaning and purpose in an environment facilitating response and interaction." The musical instruments are selected by the therapist such that the client is able to make sounds without needing prior knowledge or skill [18]. The therapeutic emphasis on interaction without prior expertise suggests that DistributeDJ could be a valuable tool both in group therapy settings as well as allowing a large variety of instrument interaction options. Its portability and flexibility also offer advantages over effective but cumbersome systems currently in use [19].

V. Summary

In this paper we presented a design of a distributed music service for group interaction. This application allows multiple users to access a common storage of musical objects that are arranged in a way that allows different juxtapositions and manipulations in a centralized global musical composition. A large benefit of using computer technology to enable such group improvisations is participant distribution: a performer need not be physically located on the stage or even in a centralized performance space to be counted as an equal participant. Related is the quality of anonymity, and a possible corresponding increased confidence in otherwise shy participants. While usually a positive contribution, it has been shown in other computer mediated communications systems, such as internet chat forums, that this protection of anonymity can sometimes lead to unwanted behavior and disruptive interactions. [17]

There are other challenging issues with using technology for group music making. Information networks do not always deliver their data with 100% accuracy. There are compromises made between guaranteeing transmission of all the information and the reliability of the delivery times. Systems with complex network configurations and a large number of participants can expect some loss of information or timing accuracy.

The performative information channels between human participants can also suffer when engaged in a computer mediated performance. Subtle non-verbal communication cues can be lost. Focus on the computer scree creates difficulty establishing eye contact, making it hard to be aware of a fellow performer's intentions.

Despite the challenges, the availability of technology capable of participating in collaborative music making makes it a attractive target for research. The number of personal computer and mobile devices is staggering and they are often within very close proximity to their human controllers, whether it be a computer in a car, tablet in a briefcase, or smart phone in a front pocket. The ubiquity of computer mediated communication devices can truly allow a participant to make collaborative music in any place and at any time. From this perspective, these devices are superseded only by the voice as the most readily available complex musical instrument.

References

[1] ActivClassroom. http://www.prometheanworld.com/upload/pdf/.

[2] Xavier Amatriain, Pau Arumi, and David Garcia. A framework for efficient and rapid development of cross-platform audio applications. *Multimedia Systems*, 2007.

[3] Nicholas J. Bryan, Jorge Herrera, Jieun Oh, and Wang Ge. Momu: A mobile music toolkit. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2010.

[4] Arshia Cont and Cyrille Henry. Real-time Gesture Mapping in Pd Environment using Neural Networks. *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04)*, pages 39–42, 2004.

[5] Miller Puckette: Pure Data. http://crca.ucsd.edu/ msp/software.html.

[6] Georg Essl. Urmus: http://urmus.eecs.umich.edu/.

[7] Rebecca Fiebrink, P.R. Cook, and Dan Trueman. Play-along mapping of musical controllers. In *Proc. International Computer Music Conference*. Citeseer, 2009.

[8] Anne Galloway, H. Sundholm, Martin Ludvigsen, and Alan Munro. From Bovine Horde to Urban Players: Multidisciplinary Interaction Design for Alternative City Tourisms. In *Workshop position paper for MUM*, pages 1–6. Citeseer, 2003.

[9] H.S. Garcia. Understanding Collective Gestural Improvisations; a Computational Approach. (August), 2006.

[10] CE. Guzzetta. Effects of relaxation and music therapy on patients in a coronary care unit with presumptive acute myocardial infarction. *Holistic Nursing Consultants*, 1989.

[11] Tom Hope, Y. Nakamura, Toru Takahashi, Atsushi Nobayashi, S. Fukuoka, M. Hamasaki, and T. Nishimura. Familial collaborations in a museum. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 1963–1972. ACM, 2009.

[12] Freeman Jason, Carl DiSalvo, Michael Nitsche, and et al. Urban remix: http://gtcmt.gatech.edu/?p=6351.

[13] Oh Jieun, Jorge Herrera, Nicholas J. Bryan, Luke Dahl, and Ge Wang. Evolving the mobile phone orchestra. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2010.

[14] Cycling74: Max/MSP. http://cycling74.com/products/maxmspjitter/.

[15] Keith Mullins, Margaret Cahill, and Computer Music. Importing MusicXML files into Max / MSP. *Information Systems*.

[16] OSCulator. http://www.osculator.net/.

[17] Helen Sharp, Yvonne Rogers, and Jenny Preece. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2007.

[18] Rachel D Smith and Helen M. Patey. *Music therapy*. Sage Publishing, 2003.

[19] Soundbeam. http://www.soundbeam.co.uk/.

[20] Lynn A. Stein. CHALLENGING THE COMPUTATIONAL METAPHOR : IMPLICATIONS FOR HOW WE THINK. *Technology*, pages 473–507, 1999.

[21] W Stevens, Bill Fenner, and Andrew M. Rudoff. Unix network programming. *Addison-Wesley Professional*, 2003.

[22] Atau Tanaka. Mobile music making. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2004.

[23] Sue Targett and Mikael Fernström. AUDIO GAMES : FUN FOR ALL ? ALL FOR FUN ? Sue Targett EIRÍ Corca Baiscinn Community Centre , Circular Road , Kilkee , Interaction Design Centre Dept . of Computer Science and Information Systems , University of Limerick , Ireland. *Proceedings of the 2003 International Conference on Auditory Display*, (July):6–9, 2003.

[24] TouchOSC. http://hexler.net/software/touchosc.

[25] Gil Weinberg, Andrew Beck, and Mark Godfrey. Zoozbeat: a gesture-based mobile music studio. *In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2009.

[26] A.D. Wilson and A.F. Bobick. Realtime online adaptive gesture recognition. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 270–275. IEEE, 2000.