BOEUF: A Unified Framework for Modeling and Designing Digital Orchestras

Florent Berthaut, Luke Dahl

BOEUF : a Unified Framework for Modeling and Designing Digital Orchestras

Florent Berthaut[1⋆] and Luke Dahl[2]

[1] Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France
[2] University of Virginia
florent@hitmuri.net   lukedahl@virginia.edu

Abstract. Orchestras of Digital Musical Instruments (DMIs) enable new musical collaboration possibilities, extending those of acoustic and electric orchestras. However the creation and development of these orchestras remain constrained. In fact, each new musical collaboration system or orchestra piece relies on a fixed number of musicians, a fixed set of instruments (often only one), and a fixed subset of possible modes of collaboration.
In this paper, we describe a unified framework that enables the design of Digital Orchestras with potentially different DMIs and an expandable set of collaboration modes. It relies on research done on analysis and classification of traditional and digital orchestras, on research in Collaborative Virtual Environments, and on interviews of musicians and composers. The BOEUF framework consists of a classification of modes of collaboration and a set of components for modelling digital orchestras. Integrating this framework into DMIs will enable advanced musical collaboration modes to be used in any digital orchestra, including spontaneous jam sessions.

Keywords: Boeuf, orchestra, collaboration, framework, digital musical instrument, digital orchestra, dmi, nime, collaborative music

1   Introduction

Orchestras of Digital Musical Instruments (DMIs) began to appear at the end of the 1970s, with the League of Automatic composer and later The Hub [8]. With the subsequent generalisation of hardware platforms, music software and communication protocols such orchestras have become more common. A Digital Orchestra (DO) can be composed of a single multi-user DMI [12]; a homogeneous collection of identical DMIs, such as in most laptop [24] and mobile phone [18] orchestras ; or a set of DMIs of different types. In turn, DMIs may rely on various hardware interaction devices, audio and visual feedback modalities, software architectures and sound synthesis techniques.

Digital orchestras enable new modes of collaboration that were not possible in ensembles of acoustic or electric instruments. For example, musicians can share audio or control streams, thus processing the output of one instrument in another instrument. They may exchange building blocks, such as note sequences or sound material. Or a musician might even allow their instrument to be controlled by another musician. Finally, collaboration modes present in traditional ensembles, such as non-verbal communication for coordinating musical cohesion or variation [21], can be augmented in DOs.

However, the design and implementation of digital orchestras remains complex, especially when they include diverse musical instruments. In fact, most existing orchestras rely on frameworks that are specific to a software or hardware platform or even to a single instrument. These can be in the form of a set of Max/MSP patches that connect together, a fixed set of hardware interconnections (e.g. MIDI cables), and so on. Even if standard communication protocols such as MIDI or Open Sound Control (OSC) are used, each orchestra manages communication between instruments in a specific way (e.g. namespaces, ports, channels), thus prohibiting new instruments from easily joining. Therefore, orchestras of mixed instruments are more complex to organise, and, as in the case of spontaneous jam sessions, advanced collaboration modes may be completely inaccessible. Finally, collaboration modes are mostly reimplemented with every new orchestra or piece, often only as a limited subset of all collaboration possibilities.

These practises impede the development of new ensembles and the creation of music that uses the new modes of collaboration that DMIs enable. Therefore the computer music community would benefit from a framework that is simple enough to be integrated into any instrument, that takes into account existing modes of collaboration, and that can be expanded by adding new ones. Such a framework would facilitate the creation of orchestras and encourage exploration of new collaboration modes. The contributions presented in this paper are twofold:

1) We present a classification of musical *modes of collaboration*. These modes were developed after conducting a literature review of research on DOs and Collaborative Virtual Environments (CVE), and interviews and discussions with composers and musicians in DOs at the SCRIME at University of Bordeaux, and at CCRMA at Stanford University. This classification, described in Section 2.1, allows for the practical analysis of musical collaboration and can be extended to include novel collaboration modes.

2) We then provide a set of *components* that constitute a model of a digital orchestra. These components, described in Section 2.2, allow for the design of orchestras that can access any of the modes of collaboration in our classification, as demonstrated in Section 2.3.

The modes of collaboration and components form the conceptual basis of the BOEUF framework. The implementation and integration of such a framework into existing DMIs, which we discuss in Section 3, will enable advanced

musical collaboration in DOs with any set of instruments, even in the context of improvised sessions.

1.1   Related work

A number of frameworks exist for describing orchestras of DMIs and collaborative instruments. Jordà [11] classifies multi-user instruments according to the number of users and whether this can be variable, the flexibility of roles, and whether users can influence others' musical output. Blaine and Fels [4] describe collaborative interfaces according to their use (focus), properties of the instrument (location, media, physical interface, musical range, level of physicality, and pathway to expert performance), and the structure of interactions between players (scalability, player interaction, directed interaction.) Hattwick and Wanderley [9] create a space of collaborative musical systems, with dimensions of texture, equality of roles, centralisation of information, the role of physical location, whether interactions are time synchronous, and whether sound production depends on more than one performer. Our framework is intended to span this space, with the exception that ours is intended to enable only synchronous (real-time) interactions. Weinberg [25] provides a historical overview of interconnected musical networks. Weinberg's examples, as well as his descriptions of intra-player interdependencies were influential on our development of the BOEUF framework.

Several protocols and software tools have been created to deal with the sharing of musical data for both single instruments and within networked orchestras. For example, Jamoma[20] and libMapper [14] both give access to the structure and parameters of networked DMIs, sometimes with features for watching and grabbing parameters. An interesting example is the Digital Orchestra Toolbox [15] which simplifies the collaborative creation and mapping of DMIs. Most of these tools in turn rely on the Open Sound Control protocol for network communication. However, while they provide all the generic sharing and mapping features required for networked musical control, these tools do not specifically cover the modes of collaboration used in DOs, and thus fail to provide a common basis for creating orchestras of mixed DMIs.

2   The BOEUF Framework

The conceptual part of the BOEUF [3] framework consists of a set of components that allow for designing any digital orchestra, and enables a set of collaboration modes, which are organised into three categories. Our framework draws on research done on CVEs. A Collaborative Virtual Environment is defined in [22] as a "computer-based, distributed, virtual space or set of places. In such places, people can meet and interact with others, with agents or with virtual objects". The challenges of CVEs are very similar to those of DOs. As with orchestras, users in CVEs need to perceive each other and to communicate within

---

[3] for *BOEUF OrchEstras Unification Framework*, *boeuf* meaning *jam session* in French

the virtual environment in order to cooperate for specific tasks. In CVEs as in DOs, this perception is often impaired by the digital mediation. However, unlike research on musical collaboration, CVE research has gone further than merely classifying collaborative applications. Practical implementation-oriented models have been proposed [26] [10], as have frameworks that aim at helping developers build CVEs [19]. Adapting CVE frameworks and models to the musical domain provides us with insights that inform our own practical collaboration framework.

| Orchestras | Cooperation | | | Communication | | | Organisation | | |
	Independent	Complementary	Concurrent	Awareness	Indication	Exchange	Nomination	Grouping	Selection
Sound Bounce	X	X		X		X	X		X
Anahata	X	X		X		X			X
Les complémentaires	X	X		X		X	X		X
Couacs	X		X	X					X
Reactable	X	X	X	X		X	X		X
LOLC	X			X	X	X			X
The Hub	X	X							X
Intellectual Improperty	X			X	X		X	X	X
JamiOki	X			X	X				X
Cobra	X	X		X	X		X	X	X

Fig. 1. Modes of collaboration in various orchestras, pieces, and multi-user instruments

2.1 Modes of collaboration

From the existing work presented above and our interviews, we extract three categories of modes of collaboration between musicians: *Cooperation*, *Communication* and *Organisation*. We consider modes that are digitally mediated, as well as those that are not usually mediated, e.g. non-verbal communication.

We now describe these modes and present examples from a selection of orchestras, multi-user instruments, and pieces. Figure 2 shows the use of these modes in each piece.

Sound Bounce [6] is a piece for the Stanford Mobile Phone Orchestra [18]. Anahata and Les Complémentaires are both electro-acoustic trios at the SCRIME.

Couacs[3] is a multi-user instrument in the form a 3D first person shooter video game. The Reactable[12] is a multi-user instrument based on a tabletop tangible user interface. LOLC[13] is an orchestra that relies on live-coding and instant messaging. The Hub[8] is one of the first digital orchestras. *Intellectual Improperty* is a piece for the Stanford Laptop Orchestra [24]. JamiOki[23] is a system for playing game pieces, providing instructions to each musician and getting feedback from them. *Cobra* is a game piece with various collaboration modes.

Cooperation modes describe the coordination of musicians' actions with respect to their instruments. We define three subcategories that correspond to different possible interconnections between musicians and their instruments. These subcategories are inspired by the CVE cooperation framework described in [16]. Cooperation modes can therefore be: **Independent** when each musician controls their own instrument or modules. **Complementary** when two or more musicians can affect the same musical output but at different levels of the audio synthesis graph, i.e. each musician controls a different sound parameter. **Concurrent** when musicians can affect the same musical output at the same level, i.e. when multiple musicians modify the same musical parameter on a single instrument.

Independent modes of cooperation exist whenever two musicians play at the same time. The cooperation in this case consists of the coordination of gestures as each musician performs their own instrument. Complementary modes of cooperation are used in many orchestras. Our interviews with musicians from Les Complémentaires and Anahata revealed that they were both using a non-mediated complementary mode of cooperation, by spreading one sound over several musicians, one playing the attack, another the sustained part and another the end of the sound. Similarly, with the Reactable, complementary cooperation occurs when two musicians manipulate different tangibles on the same audio path. Finally, concurrent modes of cooperation are less common as they imply either conflicts or games between musicians. In *Couacs* [3] for example, musicians control avatars in a musical video game and can override the parameters of other musicians by shooting at them. Concurrency handling strategies must be applied, as discussed for example in [7], such as always performing the newest action, averaging between actions of different musicians, using a physical model with different weights, or grabbing a parameter for exclusive use.

Communication modes do not directly impact the production of sound, but rather influence the actions of the musicians, in particular those who are involved in one of the cooperation modes. In most orchestras, communication modes are non-mediated and enable musical cohesion or variation [21]. However DOs often lack visibility [5], making it difficult for performers to see or understand the state of others' activities. This issue is amplified in the case of networked orchestras when musicians are not physically collocated. These difficulties can be addressed by integrating communication modes into the framework. In the field of CVE, similar communication problems have led to the use of concepts

such as Embodiment[2], i.e. using avatars to represent users, CoPresence i.e. the feeling users have of being together in the virtual environment, and Awareness, i.e. the understanding of other users' actions [1]. This concept of awareness has also been discussed for the case of DOs in [7]. Modes of communication are divided into three subcategories. **Awareness** includes all non-intentional communication, such as making musicians' activities visible to each other for the purpose of enabling synchronisation, cohesion or variations. **Indications** are intentional communicative acts such as demonstrating gestures and intentions or sending commands. **Exchange** corresponds to transfers of musical data between musicians.

An example of awareness in Couacs [3] is the use of avatars that provide information on the musical output each player is generating. Awareness is especially important in mixed instruments orchestras, when it is not clear how each musician contributes to the musical output. One example of indication is the system of text messages sent by the conductor to instrumentists in the piece *Intellectual Improperty*. They are also used in Jamioki, to guide musicians through musical games or improvised pieces. An example of exchange can be found in *Sound Bounce* where musicians use ball-throwing gestures to pass a sound process from one player to another.

Organisation modes do not have any effect on the music produced but rather impact the communication and cooperation modes. We define three organisation modes: **Nomination** consists in defining the roles of musicians within the orchestra. **Grouping** consists in defining a hierarchy of groups of instruments. **Selection** is the act of choosing a single instrument or a group in the context of cooperation or communication, e.g. selecting which musician to send an indication to.

A common example of nomination is the role of a conductor. In most orchestras, this role is fixed and the conductor has a specific interface, such as in *Intellectual Improperty*. However, roles can also be dynamically changed as described in [11]. For example, a musician from Les Complémentaires explained how they use the role of soloist to give priority to one musician at certain moments of a performance. An example of grouping is found in *Intellectual Improperty*, where the conductor can group musicians. With the selection mode, he then chooses which group he sends a message to.

2.2   Orchestra components

In order to design orchestras that enable the collaboration modes described above, our framework includes a generic model of a digital orchestra with a set of components: Session, Group, Instrument, Module, Parameter, Output, Meter, and Message.

A **session** represents an instance of a DO. It contains the instruments and the network of possible interactions between instruments.

A **group** is a set of instruments or groups. The parameters common to all instruments in a group can be grabbed and set simultaneously. Similarly, messages sent to a group are sent to all members. For example in a poly-instrument

orchestra, a group can be all the musicians with the same instrument. A default group of all instruments is always defined, thereby giving access to parameters that are common to all instruments in the session e.g. tempo or scale.

An **instrument** represents a bounded set of music-generating processes (i.e. modules) and a user interface. It has parameters, outputs, meters and it can receive and send messages. We presume that each musician in the orchestra is in control of at least one instrument. Thus, in our modeling an instrument often acts as a proxy for the musician.

A **module** is a software component that produces musical data, of audio or control type. It is composed of several parameters and outputs. Modules have a type, possibly from a common set of types (each with a predefined number of parameters and data). This way a module can be copied by another instrument if this instrument handles modules of the same type. For example, a module of type LowPassFilter might have *Cutoff Frequency* and *Q* as parameters, whereas a module of type *MidiPattern* will hold an array of MIDI events. Many instruments have an internal structure that is more complex than a simple chain of modules. They may have complex audio graphs with many hierarchical levels or feedback. However, from the perspective of user interaction these can usually be flattened to a set of modules and associated parameters .

A **parameter** is an attribute of a module or instrument that influences its musical production. Parameters can be of various types such as MIDI events, float or integer values, input audio streams, and so on. Parameters can be:

 − *Retrieved*: the current value of the parameter is returned once.
 − *Watched*: the value of the parameter is sent every time it changes, until it is not watched anymore.
 − *Indicated*: a new value is proposed for the parameter but not set.
 − *Set*: the parameter is set to a new value.
 − *Grabbed*: the parameter can only be set by the instrument that grabbed it.

These actions are always accessible to the instrument that owns the parameter, but authorisation might be needed for other instruments to access a parameter. Concurrent access can be managed in different ways, the simplest being by grabbing a parameter so that only one instrument can access it.

An **output** is a musical attribute that is produced by a module or instrument. Outputs can be of the same types as parameters. Outputs can be retrieved and watched by another instrument without requiring authorisation.

A **meter** is a component of an instrument that is not used in the actual sound production, but rather gives an indication on the activity of the instrument, e.g. spectrum or loudness.

A **message** is a text, image, or video sent from one instrument to another instrument or group. Messages can be standard (e.g. Start, Stop, Fade Out), defined per session instruments, or dynamically created.

2.3   Modelling Digital Orchestras with BOEUF

The BOEUF framework allows for the analysis and design of digital orchestras with respect to the collaboration modes described in Section 2.1, using the set of components defined in Section 2.2.
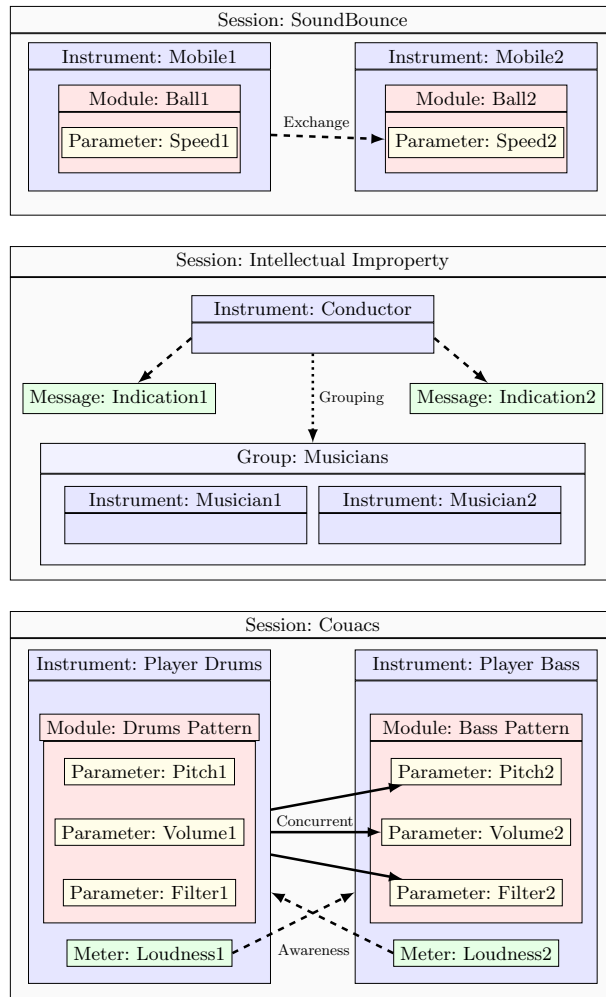
Fig. 2. Designing orchestras using our framework. Dashed lines are used for Communication, solid lines for Cooperation, and dotted lines for Organisation. From top to bottom: Sound Bounce, Intellectual Improperty, Couacs

In this section we discuss how one can implement each mode of collaboration using the components. Where applicable, we will illustrate the process by referring to three examples: *Sound Bounce*, *Couacs* and *Intellectual Improperty*. We do not claim that these examples were designed with our framework in mind. Rather, we show how they could be implemented using the BOEUF components. This amounts to a form of analysis. A graphical representation of these analyses can be seen in Figure 2.

Cooperation modes are handled by the access each instrument has to the parameters and outputs of other instruments. Independent cooperation is achieved simply by having each instrument belong to the same session.

Complementary cooperation is achieved when one musician, through his or her instrument, has permission to grab or set a parameter on another musician's instrument, and that parameter is not currently under control by the owner of that instrument. Another way to achieve complementary cooperation would be for one instrument to retrieve or watch the output of another musician's module or instrument, and process that output.

Concurrent cooperation occurs when two musicians (through their instruments) have access to set the same parameter on an instrument. Different strategies of concurrency handling can be defined by each instrument for each parameter. In Couacs concurrent cooperation involves one player (instrument) setting all the parameters of the module of another player at once by shooting this player.

Communication modes are implemented through various components The most basic form of awareness is achieved when musicians can hear the musical output of each others' instruments. The possibility for awareness is enhanced when one musician (through their instrument) can watch a parameter, output, or meter on another musician's instrument. In Couacs the size of each player's avatar is scaled according to the current loudness of their instrument, allowing the musicians to perceive who is playing what. This can be achieved in our framework by adding a loudness meter to each instrument, and having each instrument watch the meters of the other instruments. The details of the GUI (e.g. in Couacs, avatars that change size) are not specified in BOEUF.

Indication can take the form of messages or commands from one instrument to another. This can be as simple as one instrument indicating a new value for a parameter on another's instrument. (It is up to the implementation to decide how to display this indication on the receiver's GUI.) In *Intellectual Improperty*, indication takes place when the conductor sends messages with instructions to various groups of musicians. Communicating by indication may also take place when a musician demonstrates a musical gesture, for example when they want other musicians to imitate them. This form of indication is enabled when various means of awareness are implemented, and can be achieved in a number of ways. For example, if some musicians are watching a parameter on the indicator's

instrument, they may notice how this parameter changes when the indicator performs a demonstrative sound.

Exchange can be performed by copying modules or sets of parameters. For example, in *Sound Bounce*, when one player throws a soundball to another player, the values of all parameters in the thrower's Ball module are copied into the parameters of the receiver's Ball module.

Organisation modes are mostly implemented through the group and instrument components. Nomination of the conductor in *Intellectual Improperty* corresponds to the defining of a group containing only the conductor. The conductor can also define groups (grouping) to which they add (nomination) musicians (by proxy of their instruments).

An example of selection can be found in *Sound Bounce*, when a player aims their phone at another player to whom they wish to throw their soundball. In our framework, the player's instrument selects another instrument from a group in order to perform an exchange.

3   Discussion

We have described the BOEUF framework, which consists of the modes of collaboration that digital orchestras should have available to them, and a set of components which model digital orchestras and enable these modes. In this section we anticipate challenges that may arise when integrating our framework into new and existing DMIs. First we discuss issues related to implementation, and secondly we look at aspects related to interfaces for accessing BOEUF's features.

3.1   Integration

In order to ensure portability, the low level layer of the BOEUF implementation should rely on a commonly used network protocol such as Open Sound Control. Depending on the type of DMI, we envisage four levels of integration of BOEUF, each with progressively greater functionality.

Hardware Box: In the case of black-box instruments (i.e. hardware synthesisers and instruments with fixed software structure), BOEUF integration would require a dedicated hardware device. This device would connect to the instrument via MIDI, audio and OSC input and output ports. The device would in turn connect to the BOEUF session via a network connection. It would provide a limited set of modes of collaboration with a limited number of instruments. This limitation depends on the number of ports, controls and displays available on the device.

Software Plugin / Standalone: In the case of DMIs implemented as non-reprogrammable software components, a BOEUF software plugin or standalone application can be connected to the virtual audio and control ports of the DMI. The BOEUF software would in turn connect to the BOEUF session. Compared to a BOEUF hardware device, the software application can enable more modes of

collaboration. In particular, thanks to the software's graphical interface, aware-
ness and indication modes of communication, as well as cooperation and organ-
isation modes with a flexible number of other instruments, are now possible.

Open Sound Control Namespace: In the case of an instrument with pro-
grammable input and output layers, a BOEUF session can be accessed directly
via Open Sound Control messages following the BOEUF protocol. The input
layer must parse incoming messages, send them to the appropriate components
of the instrument, and perform any required actions (e.g. set, grab, or retrieve
the value of a parameter). The output layer must emit messages to other instru-
ments, for example when the value of a watched parameter is changed, or when
the user wants to send a message to another instrument or group of instruments.
This integration would resemble that of the Jamoma framework in Max/MSP.

Software Library: Finally, in the case of open-source software instruments,
a BOEUF software library can be integrated into the source code of the instru-
ment. This library would manage the network connection to the session inter-
nally. The instrument must register its components with the BOEUF library and
implement callback functions that provide access to its components.

3.2   Interfaces

The new modes of collaboration enabled by digital orchestras and described
in our framework generate new streams of information that a musician must
apprehend and respond to, and they create new opportunities for action. These
lead to two challenges in the development of a DMI interface that would give
access to these collaboration modes.

The first concerns the visualisation of information available through the
BOEUF framework. In order to enable all modes of collaboration, an inter-
faces should provide ways of displaying information from various components in
the framework, such as meters for awareness, messages for indication, modules
for exchange modes, and so on. However, we also want to preserve non-mediated
modes of collaboration such as the non-verbal communication the occurs between
musicians who can see each other. If a DMI interface disrupts the perception a
musician has of the other musicians in the orchestra, it might degrade rather
than enhance the collaboration. For instance, meters can increase awareness of
what each musician is playing, but only if they are easily associated with their
respective musicians and their gestures.

The second challenge relates to the new actions that are made available in
the modes of collaboration and through the components of BOEUF. Controls for
these actions might be provided on a dedicated interface, such as a control surface
or touchscreen. Or, in the case of setting a parameter on another instrument,
the parameter under control might be temporarily mapped to sensors on the
musician's own instrument. In the case of instruments with a graphical interface,
representations of the components can be integrated as part of the interface of
the instrument. However, these controls should not interfere with the interface of
the instrument itself, for example by reducing the number of available sensors or
possibilities for musical expression possibilities, or by disrupting the musician's
perception.

One way to achieve these objectives might be through the use of Augmented Reality displays, such as those described in [17]. Semi-transparent mirrors and displays are combined to visually augment a physical space. The augmentations are only seen when looking at that space through the mirrors, either from only one or from both sides of them. This would allow for:

- Co-located visualisation of parameters, meters and messages with the physical musicians and instruments, e.g. displaying meters and messages overlapping musicians.
- Access to parameters of other instruments by overlapping a 3D representation of them with the physical interface.
- Individual views of the components depending on the role of the musician, so that a conductor can for example see the separation of musicians in groups of instruments.

When designing interfaces that increase the musicians' access to information, we must also consider how the interface affects the ability of the audience to perceive and understand the musicians' actions. This visibility is important in that it allows the audience to perceive the risk inherent in the situation, and thus know that they are co-participants in a valid musical performance [5].

To that extent, audience members can be given access to the BOEUF interface. Information from the BOEUF framework would be added to the usual perception of musicians' performance, and would help spectators understand both the interactions between musicians and the contribution of each musician to the music produced by the orchestra.

4   Conclusion

In this paper we introduced BOEUF, a framework for modeling, designing, building, and managing orchestras of DMIs. It consists of a classification of possible collaboration modes, and of a set of components that can be used to implement them. We demonstrated how BOEUF can be used for modelling and designing orchestras with three examples: a piece for mobile phone orchestra, a multi-user instrument, and a piece for laptop orchestra.

Our next steps are: to investigate the integration of the BOEUF framework into DMIs, following the guidelines described in Section 3; to design novel interfaces for accessing BOEUF, possibly using Augmented Reality displays as proposed in Section 3; to study the impact of BOEUF on musicians' collaboration in the context of spontaneous orchestras or jam sessions; and to use our framework for in-depth investigations into issues such as concurrent access to musical parameters, indication of musical gestures through parameters, and visualisation of musical activity with meters.

We believe that this research is essential in that it will empower the community to more easily explore new musical collaboration possibilities.

References

1. Benford, S., Bowers, J., Fahlén, L.E., Greenhalgh, C.: Managing mutual awareness in collaborative virtual environments. In: Proceedings of VRST. pp. 223–236 (1994)
2. Benford, S., Bowers, J., Fahlén, L.E., Greenhalgh, C., Snowdon, D.: User embodiment in collaborative virtual environments. In: Proceedings of CHI 95. pp. 242–249. New York, NY, USA (1995)
3. Berthaut, F., Katayose, H., Wakama, H., Totani, N., Sato, Y.: First Person Shooters as Collaborative Multiprocess Instruments. In: Proceedings of NIME 11. pp. 44–47. Oslo, Norway (2011)
4. Blaine, T., Fels, S.: Contexts of collaborative musical experiences. In: Proceedings of the 2003 Conference on New Interfaces for Musical Expression. pp. 129–134. NIME '03, National University of Singapore, Singapore, Singapore (2003), `http://dl.acm.org/citation.cfm?id=1085714.1085745`
5. Dahl, L.: Wicked problems and design considerations in composing for laptop orchestra. In: Proceedings of NIME 12 (2012)
6. Dahl, L., Wang, G.: Sound bounce: Physical metaphors in designing mobile music performance. In: Proceedings of NIME 10, Sydney, Australia (2010)
7. Fencott, R., Bryan-Kinns, N.: Hey Man, you're invading my Personal Space! Privacy and Awareness in Collaborative Music. In: Proceedings of NIME 10. pp. 198–203 (2010), `http://www.educ.dab.uts.edu.au/nime/PROCEEDINGS/papers/Paper\%20J1-J5/P198\_Fencott.pdf`
8. Gresham-Lancaster, S.: The aesthetics and history of the hub: The effects of changing technology on network computer music. Leonardo Music Journal pp. 39–44 (1998)
9. Hattwick, I., Wanderley, M.M.: A dimension space for evaluating collaborative musical performance systems (2012)
10. Hindmarsh, J., Fraser, M., Heath, C., Benford, S., Greenhalgh, C.: Object-focused interaction in collaborative virtual environments. ACM Trans. Comput.-Hum. Interact. 7(4), 477–509 (Dec 2000), `http://doi.acm.org/10.1145/365058.365088`
11. Jordà, S.: Multi-user instruments: models, examples and promises. In: Proceedings of NIME 05. pp. 23–26. Singapore, Singapore (2005), `http://dl.acm.org/citation.cfm?id=1085939.1085948`
12. Jordà, S., Kaltenbrunner, M., Geiger, G., Bencina, R.: The reactable*. In: Proceedings of the International Computer Music Conference (2005)
13. Lee, S.W., Freeman, J., Colella, A., Yao, S., Van Troyer, A.: Collaborative musical improvisation in a laptop ensemble with lolc. In: Proceedings of the 8th ACM conference on Creativity and cognition. pp. 361–362. C&C '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2069618.2069696`
14. Malloch, J., Sinclair, S., Wanderley, M.M.: Libmapper:(a library for connecting things). In: CHI'13 Extended Abstracts on Human Factors in Computing Systems. pp. 3087–3090. ACM (2013)
15. Malloch, J., Sinclair, S., Wanderley, M.: A network-based framework for collaborative development and performance of digital musical instruments. In: Computer Music Modeling and Retrieval. Sense of Sounds, Lecture Notes in Computer Science, vol. 4969, pp. 401–425 (2008), `http://dx.doi.org/10.1007/978-3-540-85035-9_28`
16. Margery, D., Arnaldi, B., Plouzeau, N.: A general framework for cooperative manipulation in virtual environments. In: In Proc. of EGVE 99: Eurographics Workshop on Virtual Environments. pp. 169–178. Springer (1999)

17. Martinez Plasencia, D., Berthaut, F., Karnik, A., Subramanian, S.: Through the combining glass. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology. pp. 341–350. UIST '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2642918.2647351
18. Oh, J., Herrera, J., Bryan, N.J., Dahl, L., Wang, G.: Evolving the mobile phone orchestra. In: Proceedings of NIME 10. Sydney, Australia (2010)
19. Pinho, M.S., Bowman, D.A., Freitas, C.M.: Cooperative object manipulation in immersive virtual environments: framework and techniques. In: Proceedings of the ACM symposium on Virtual reality software and technology. pp. 171–178. VRST '02, ACM, New York, NY, USA (2002), http://doi.acm.org/10.1145/585740.585769
20. Place, T., Lossius, T.: Jamoma: A modular standard for structuring patches in max. In: Proceedings of the International Computer Music Conference. pp. 143–146 (2006)
21. Seddon, F., Biasutti, M.: A comparison of modes of communication between members of a string quartet and a jazz sextet. Psychology of Music 37(4), 395–415 (2009), http://pom.sagepub.com/content/37/4/395.abstract
22. Snowdon, D.N., Munro, A.J.: Collaborative Virtual Environments: Digital Places and Spaces for Interaction. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2001)
23. Vigoda, B., Merrill, D.: Jamioki-purejoy: a game engine and instrument for electronically-mediated musical improvisation. In: Proceedings of the 7th international conference on New interfaces for musical expression. pp. 321–326. NIME '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1279740.1279810
24. Wang, G., Bryan, N., Oh, J., Hamilton, R.: Stanford laptop orchestra(slork). In: Proceedings of the International Computer Music Conference. pp. 505–508 (2009)
25. Weinberg, G.: Interconnected musical networks: Toward a theoretical framework. Comput. Music J. 29(2), 23–39 (Jun 2005), http://dx.doi.org/10.1162/0148926054094350
26. Wolff, R., Roberts, D.J., Steed, A., Otto, O.: A review of telecollaboration technologies with respect to closely coupled collaboration. Int. J. Comput. Appl. Technol. 29(1), 11–26 (Jun 2007), http://dx.doi.org/10.1504/IJCAT.2007.014056