BOEUF : a Unified Framework for Modeling and Designing Digital Orchestras

Florent Berthaut[1⋆] and Luke Dahl[2]

[1] University of Bristol
[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. Implementing this framework as a software library will enable advanced musical collaboration modes to be used in any orchestra, even spontaneous ones.

Keywords: Boeuf, orchestra, collaboration, digital musical instrument, CVE

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 [7]. With the subsequent generalization of hardware platforms, music software and communication protocols, the number of orchestras has grown. A Digital Orchestra (DO) can be composed of a single multi-user DMI [10], or of multiple DMIs of the same type, such as in most laptop [17] or mobile phones [14] orchestras, or of a set of DMIs of different types. In turn, the DMIs may rely on various hardware interaction devices, feedback modalities, software architectures and sound synthesis techniques. DOs enable new modes of collaboration that were not possible with acoustic or electric orchestras. For example, musicians can share audio or control streams, thus processing the output of one instrument in another instrument. They may exchange precomposed building blocks, such as patterns or sounds. Or a musician might even allow their instrument to be controlled by

[⋆] This project was partially funded through the Marie Curie FP7 framework (Grant Agreement PIEF-GA-2012-330770).

another musician. Finally, collaboration modes present in traditional ensembles, such as non-verbal communication for coordinating musical cohesion or variation [15], 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 specific to a software or hardware platform or even to a single instrument. Even if standard communication protocols such as MIDI or OpenSoundControl are used, each orchestra manages communication between instruments in a specific way, thus prohibiting new instruments from easily joining. Thus, orchestras of mixed instruments are more complex to build, and, as in the case of spontaneous jam sessions, advanced collaboration possibilities may be completely inaccessible. Often these orchestras rely on a limited set of collaboration possibilities, which are reimplemented for each new orchestra or musical piece.

These practices result in a number of impediments to the exploration of musical collaborations using DMIs. 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 collaboration modes. 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* based on a literature review of research on DOs and Collaborative Virtual Environments (CVE), and on interviews and discussions we conducted 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 compose a Digital Orchestra. These components, described in section 2.2, allow for the design of orchestras with access to any mode of collaboration in our classification, as demonstrated in section 2.3.

These form the conceptual basis of the BOEUF framework. Their implementation in the form of a software library, and the integration of this library in various software components 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 systems exist for describing orchestras of DMIs and collaborative instruments. Jorda [9] 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 [3] 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 [8] create a space of collaborative musical systems, with dimensions of texture, equality of roles, centralization 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 [18] 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, Minuit, Jamoma and libMapper all 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 [12] which simplifies the collaborative creation and mapping of DMIs. Most of these tools in turn rely on the OpenSoundControl 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 [16] 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". 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 the virtual environment in order to cooperate for specific tasks. However, unlike research on musical collaboration, CVE research has gone further than merely classifying collaborative applications. Practical implementation-oriented models have been proposed, as have frameworks that aim at helping developers build CVEs. Adapting CVE frameworks and models to the musical domain provides us with insights that inform our own practical collaboration framework.

2.1 Modes of collaboration

From the existing work presented above and our interviews, we extract three modes of collaboration between musicians: *Cooperation*, *Communication* and *Organisation*. Figure 1 shows the use of these modes in a selection of orchestras and multi-user instruments and pieces. *Sound Bounce*[5] is a piece for the Stanford Mobile Phone Orchestra [14]. *Anahata* and *Les complementaires* are both

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

Orchestra/Instrument	Cooperation			Communication			Organisation		
	Independent	Complementary	Concurrent	Awareness	Indications	Exchanges	Nomination	Grouping	Selection
Mopho: Sound Bounce	X	X		X		X	X		X
Anahata	X	X		X		X			X
Les complementaires	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
Slork: 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 and multi-user instruments

electro-acoustic trios at the SCRIME. *Couacs*[2] is a multi-user instrument in the form a 3D first person shooter video game. The *Reactable* is a multi-user instrument based on a tabletop tangible user interface. *LOLC[11]* is an orchestra that relies on live-coding and instant messaging. *The Hub* is one of the first Digital Orchestras. *Slork: Intellectual Improperty* is a piece for the Stanford Laptop Orchestra [17]. *JamiOki* 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 [13]. Cooperation modes can therefore be: **Independent** when multiple musicians do not control the musical output of the same instrument or module. **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. 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 Complementaires* 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, in 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* [2] 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 [6], such as applying 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 [15]. However DOs often lack visibility [4], 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, 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 [6]. 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. **Exchanges** include transfers between musicians of musical data. An example of awareness in Couacs [2] 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 indications 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 [9]. For example, a musician from Les Complementaires 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 the group he sends a message to (indication).

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 the group can be grabbed and set simultaneously. Similarly messages that are sent to a group are sent to all members. For example a group can be all the musicians with the same instrument in a poly-instrument orchestra. A default group of all instruments is always defined, thereby giving access to a list of parameters common to all instruments of 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 also has parameters, outputs, meters and it receives and sends messages. 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 set of modules. They may have complex audio graphs with many hierarchical levels. However, for the purpose of user interaction these can usually be flattened as 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, watched, indicated, set and grabbed. These actions are always accessible to the instrument that owns the parameter. Other instruments in the session can request authorization for each of these actions. 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 and that can be retrieved and watched by another instrument without authorization. Outputs can be of the same types as parameters. 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, picture 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 Designing Orchestras using BOEUF

The BOEUF framework allows for the design of orchestras with all collaboration modes described in section 2.1, using the set of components that we defined in section 2.2. Let us illustrate that process with three examples: *Sound Bounce*, *Couacs* and *Intellectual Improperty.* A graphical representation of their design can be seen in Figure 2. Cooperation modes are handled by the access each instrument has to the parameters and outputs of other instruments. In the case of concurrent cooperation, as explained in section 2.1, different strategies of concurrency handling can be defined by each instrument for each parameter. In

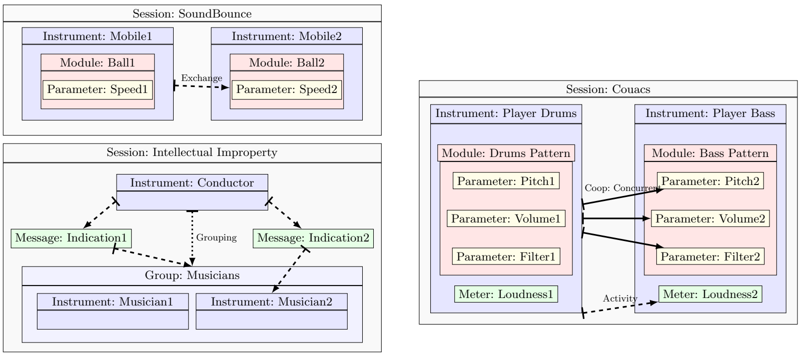

Fig. 2. Designing orchestras using our framework. Dashed lines are used for Communication, solid lines for Cooperation, and dotted lines for Organisation.

Couacs concurrent cooperation involves one instrument (player) setting all the parameters of the module of another player at once by shooting this player. Communication modes are implemented through various components. Awareness is enabled by the use of meters and watched parameters and outputs. In *Couacs*, the players' avatars are scaled using the loudness meters of each instrument, allowing musicians to perceive who is playing what. Indications rely on the use of messages and indicated parameters. In *Intellectual Improperty*, messages are sent by the conductor to the musicians.

Exchanges are performed by copying modules or sets of parameters. For example, in *Sound Bounce*, modules representing sound balls are sent from one instrument to another. Organisation modes are mostly implemented through the group and instrument components. Nomination of the conductor in *Intellectual Improperty*, corresponds to the definition of a group for the conductor only. The conductor can define (nominate) groups of musicians (instruments) to send indications (messages) to. An example of selection can be found in *Sound Bounce*, where each smartphone (instrument) selects another instrument for the exchange of sound balls (modules).

3 Conclusion

In this paper we introduced BOEUF, a framework for designing, building, and managing orchestras of DMIs. Our next step is to implement this framework as a software library which can be integrated into DMIs. This will facilitate the modes of collaboration described here, and empower the community to more easily explore new musical collaboration possibilities.

Another research area raised by BOEUF has to do with questions of usability when accessing BOEUF components during performance in order to enable or enact the various modes of collaboration. For example, we would like to study-how to best enable a musician to grant another musician complementary or concurrent access to a parameter on their own instrument.

References

1. S. Benford, J. Bowers, L. E. Fahlén, and C. Greenhalgh. Managing mutual awareness in collaborative virtual environments. In *Proceedings of VRST*, pages 223–236, 1994.
2. F. Berthaut, H. Katayose, H. Wakama, N. Totani, and Y. Sato. First Person Shooters as Collaborative Multiprocess Instruments. In *Proceedings of NIME 11*, pages 44–47, Oslo, Norway, 2011.
3. T. Blaine and S. Fels. Contexts of collaborative musical experiences. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, NIME '03, pages 129–134, Singapore, Singapore, 2003. National University of Singapore.
4. L. Dahl. Wicked problems and design considerations in composing for laptop orchestra. In *Proceedings of NIME 12*, 2012.
5. L. Dahl and G. Wang. Sound bounce: Physical metaphors in designing mobile music performance. In *Proceedings of NIME 10, Sydney, Australia*, 2010.
6. R. Fencott and N. Bryan-Kinns. Hey Man, you're invading my Personal Space! Privacy and Awareness in Collaborative Music. In *Proceedings of NIME 10*, pages 198–203, 2010.
7. S. Gresham-Lancaster. The aesthetics and history of the hub: The effects of changing technology on network computer music. *Leonardo Music Journal*, pages 39–44, 1998.
8. I. Hattwick and M. M. Wanderley. A dimension space for evaluating collaborative musical performance systems, 2012.
9. S. Jordà. Multi-user instruments: models, examples and promises. In *Proceedings of NIME 05*, pages 23–26, Singapore, Singapore, 2005.
10. S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reactable*. In *Proceedings of the International Computer Music Conference*, 2005.
11. S. W. Lee, J. Freeman, A. Colella, S. Yao, and A. Van Troyer. Collaborative musical improvisation in a laptop ensemble with lolc. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, pages 361–362, New York, NY, USA, 2011. ACM.
12. J. Malloch, S. Sinclair, and M. Wanderley. A network-based framework for collaborative development and performance of digital musical instruments. In *Computer Music Modeling and Retrieval. Sense of Sounds*, volume 4969 of *Lecture Notes in Computer Science*, pages 401–425. 2008.
13. D. Margery, B. Arnaldi, and N. Plouzeau. A general framework for cooperative manipulation in virtual environments. In *In Proc. of EGVE 99: Eurographics Workshop on Virtual Environments*, pages 169–178. Springer, 1999.
14. J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang. Evolving the mobile phone orchestra. In *Proceedings of NIME 10*, Sydney, Australia, 2010.
15. F. Seddon and M. Biasutti. A comparison of modes of communication between members of a string quartet and a jazz sextet. *Psychology of Music*, 37(4):395–415, 2009.
16. D. N. Snowdon and A. J. Munro. *Collaborative Virtual Environments: Digital Places and Spaces for Interaction.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
17. G. Wang, N. Bryan, J. Oh, and R. Hamilton. Stanford laptop orchestra(slork). In *Proceedings of the International Computer Music Conference*, pages 505–508, 2009.
18. G. Weinberg. Interconnected musical networks: Toward a theoretical framework. *Comput. Music J.*, 29(2):23–39, June 2005.