ISEA2015
Proceedings of the 21st International Symposium on Electronic Art
ISSN: 2451-8611
ISBN: 978-1-910172-00-1

Manifesto for a Musebot Ensemble: A platform for live interactive performance between multiple autonomous musical agents

Oliver Bown[1], Benjamin Carey[2], Arne Eigenfeldt[3]

[1]Design Lab, University of Sydney,
[2]Creativity and Cognition Studios, University of Technology Sydney,
NSW, Australia
[3]School of Contemporary Arts, Simon Fraser University, Vancouver, BC, Canada.
oliver.bown@sydney.edu.au, Benjamin.Carey@uts.edu.au, arne_e@sfu.ca

Abstract

In this paper we draw on previous research in musical metacreation (MuMe) to propose that novel creative forms are needed to propel innovation in autonomous creative musical agents. We propose the "musebot", and the "musebot ensemble", as one such novel form that we argue will provide new opportunities for artistic practitioners working in the MuMe field to better collaborate, evaluate work, and make meaningful contributions both creatively and technically. We give details of our specification and designs for the musebot ensemble project.

Keywords

Generative music, autonomous agents, performance, computer music, musical metacreation, live algorithms.

Introduction

Musical metacreation (MuMe)[1] is an emerging term describing the body of research concerned with the automation of any or all aspects of musical creativity. It looks to bring together and build upon existing academic fields such as algorithmic composition [1], generative music [2], machine musicianship [3] and live algorithms [4]. The wider field of metacreation [5] involves using tools and techniques from artificial intelligence, artificial life, and machine learning, themselves often inspired by cognitive and life sciences. MuMe suggests exciting new opportunities for creative music making: discovery and exploration of novel musical styles and content, collaboration between human performers and creative software partners, and design of systems in gaming, entertainment and other experiences that dynamically generate or modify music.

In this paper, we begin by making the case that there is a need to establish new contexts (in effect, new 'genres') that allow incremental innovation in musical metacreation. To this effect, we present our recent efforts to design and build the infrastructure necessary to bring together community-created software agents in multi-agent performances. We frame the current proposal in its social and technical context, make a case for the value of such a project and the opportunities it will bring, discuss the challenges and questions faced, and present the design and specification of our multi-agent system, along with a set of tools and example agents that we have created.

Objectives and 'Genres' of Musical Metacreation

MuMe straddles and sometimes integrates scientific and artistic objectives. Some MuMe tasks have identifiable measures of success — either because they are fully objective [6], or can be clearly measured by expert users [7]. Others have clear usability goals in contexts where the aim is to support creativity [8]. Yet others face problems of evaluation because of their creatively open-ended nature [9]. As an example, the established practice of creating autonomous software agents for free improvised performance [10] usually involves idiosyncratic, non-idiomatic systems, created by artist-programmers [11, 12]. A recent paper by the authors [13] discussed how curators of MuMe concerts face a challenge in balancing aesthetic and technical factors in their evaluation of submitted works. The paper also showed how evaluating the degree of autonomy in systems is non-trivial and involves detailed discussion and analysis, including subjective factors. The paper identified the gradual emergence of MuMe specific genres — i.e., sets of aesthetic and social conventions — within which meaningful questions of relevance to MuMe research could be further explored. We posited that through the exploration of experimental MuMe genres we could create novel but clear creative and technical challenges against which MuMe practitioners could measure progress.

One potential MuMe 'genre' that we considered involves spontaneous performance by autonomous musical agents interacting with one-another in a software-only ensemble, created collaboratively by multiple practitioners. This concept was touted in discussions amongst MuMe practitioners, and while there are isolated instances of MuMe software agents being set up to play with other MuMe software agents, this has never been seriously developed as a collaborative project. The ongoing growth of a community of practice around generative music systems leads us to believe that enabling multi-agent performances will support new forms of innovation in MuMe research

[1] http://www.metacreation.net/mume/

and open up exciting new interactive and creative possibilities.

Previous Work

Such multi-agent "metacreative" musical performance encounters issues of network music, new models of electroacoustic ensemble performance, creative musical agents, and autonomous machine performance, all of which are well-established areas of research. In this article, we offer brief overviews of each of these research topics, in order to frame this new research area.

Network Music

With the advent of microcomputers such as the KIM-1 [14] in the 1970s, and their adoption for use by electroacoustic music composer-performers, the potential to connect them through available serial networks in concert was explored by the League of Automatic Music Composers [15]. This ensemble later grew into "The Hub", which specialized in network computer music performance [16]. As computer networks became more ubiquitous in the 2000s, their use in music similarly expanded. Weinberg [17] gives a history of network music up to 2002, and Renaud *et al.* provide a more recent review [18]. Open Sound Control (OSC) [19] has emerged as a simple and widely used format for network communication, with implementations in all commonly used computer music environments. A number of developers, particularly those working in laptop ensembles [20], have provided tools to ease network communication in the context of music collaboration (e.g., [21], [22]) and these are now emerging in commercial contexts. Widely used network-audio tools such as JackTrip [23] are evidence of the growth in practice of telematic music performance via audio communication. With the browser emerging as a veritable computer music platform under the WebAudio standard [24], network music is destined to become increasingly fluid [25].

Creative network music practitioners such as laptop orchestra organisers have, through their work, thoroughly explored those musical aesthetic considerations that may contribute to an idea of what we might expect from a multi-agent metacreative performance.

Laptop Performance

The move from analogue to digital technologies in the 1990s provided composers with many new tools. For many, the computer replaced the traditional analogue studio as a music production facility. The laptop became a vehicle of solo music performance [26] and live generative music [27], and afforded new models for digital orchestras [28]. The popularity of the laptop orchestra has generated many new compositional models [29, 30, 31] and tools [32]. Laptop performance introduces many possible forms of automated control of realtime music, such as timing quantised transitions, executing envelopes, choosing random numbers and more responsive behaviours such as pitch tracking. These are typically seen as too weak to be described as 'autonomous' behaviours, although any strict measure of autonomy is elusive. Importantly, the growing power, programmability and sheer diversity of laptop performance styles and techniques is heralding an explosion in both generative and collaborative music techniques, available to a growing community of electronic music practitioners, including artist programmers [33]. A number of musical styles now routinely involve generative processes in their production and performance [34].

Creative Musical Agents and Autonomous Machine Performance

The potential for the application of both agent-based computing [35] and autonomous machine performance to live musical interaction has been examined in detail [36, 37, 38, 39, 40]. Bown *et al.* [41] review the history and potential for autonomous machine performance with particular attention to how artists conceptualise and work with autonomous behaviours in their practice. Live MuMe work has been largely focused in either improvised genres or classical music performance where there is generally some attempt at imitating or creating a software substitute for human performance, for which there is a rich body of literature. We are also beginning to see subtle forms of musical generativity being employed in an increasing diversity of application domains, such as in the modification of everyday experiences[2], and exotic applications such as "food opera" [42]. Commercial generative music products are emerging [43], and an increasing number of artists are moving from static recordings to forms of music distribution that may enable interactivity or generativity [44].

The "Musebot Ensemble" Proposition

A *musebot* is here defined as a "piece of software that *autonomously creates* music collaboratively with other musebots". Our project is concerned with putting together musebot ensembles, consisting of community-created musebots, and setting them up as ongoing autonomous musical installations.

The creation of intelligent music performance software has been predominantly associated with simulating human behaviour (e.g., [45]). However, a parallel strand of research has shed the human reference point to look more constructively at how software agents can be used to autonomously perform or create music. Regardless of whether they actually simulate human approaches to performing music [46], such approaches are more general issues of

[2] e.g., http://reactifymusic.com/portfolio/vw-underworld-play-the-road/

software performativity and agency in creative contexts [47]. The concept of a "musebot ensemble" is couched in this view. i.e., it can be understood as a new musical form which does not necessarily take its precedent from a human ensemble. We imagine that a musebot ensemble might be most naturally found in the chill-out room of a club, where attentive awareness of any performer would be minimised. We follow an open-source philosophy, allowing for anyone who is interested to offer contributions to the design of our framework, to fork the project, and to adapt it to their needs. We expect the framework to take shape through this process. As for whether individual musebot contributions are open-source is up to their makers, but we encourage it.

Our initial steps in this process include coming up how musebots should be made and controlled so that combining them in musebot ensembles is feasible, and has predictable results for musebot makers and musebot ensemble organisers. Musebots needn't necessarily exhibit high levels of creative autonomy, although this is one of the things we hope and expect they will do. Instead, the current focus is on enabling agents to work together, complement each other, and contributes to collective creative outcomes, in other words, good music.

This defines a technological challenge that, although intuitive and easy to state, hasn't to our knowledge been successfully set out before in a way that can be worked on collaboratively. For example, Blackwell and Young provided a framework for practitioners to work collaboratively on modular tools to create live algorithms [48, 49], but little community consensus was established for what interfaces should exist between modules, nor was there a suitably compelling common framework under which practitioners could agree to work. In our case, the modules correspond clearly to the instrumentation in a piece of music, and the context is more amenable to individuals working in their preferred development environment.

In order for musebots to make music together, some basic conditions need to be established: most obviously the agents must be able to listen to each other and respond accordingly. However, since we do not limit musebot interaction to human modes of interaction, we do not require that they communicate with each other only via human senses; digital symbolic communication (i.e., network messaging) has the potential to provide much more useful information about what musebots are doing, how they are internally representing musical information, or what they are planning to do. Devising a specification for musebot communication is a key strand of this research and will be discussed below. Following the community-driven approach that we advocate, we remain open to the myriad ways in which parties might choose to structure musebot communication, imposing only a minimal set of strict requirements, and offering a number of optional, largely utilitarian concepts for structuring interaction.

Motivation and Inspiration

The initial practical motivation for establishing a musebot ensemble was as a way to expand the range of genres presented at MuMe musical events. To date, our own MuMe events have focused heavily on free improvised duets between human instrumental musicians and software agents. This format has been widely explored by a large number of practitioners[3]. However, continuing with this established tradition might run the risk of stylistically pigeonholing MuMe activity.

Three possible reasons for the success of the free improvised duets genre are:

- It is non-idiomatic (insofar as it is possible to be so). More accurately, this is an area where minimal adherence to a given set of rules is central[4], as compared to more prescribed improvisation contexts such as be-bop or blues. There is a large free improvised music community for whom there is a familiar and relatively free set of expectations for engagement. This is also important as it gives the creator freedom to work with digitally created sound and gesture that does not resemble existing human musical instruments, as simulating human instrumental performance is challenging. Arguably, it is also simply easier to model: Our perception of "poor melody", "awkward harmony", and "stiff rhythms" have been set through hundreds of years of common practice music;
- One musician playing with one software agent means that there is a clear and focused process of interaction between the two elements. This can be clearly observed in most cases (except, for example, when the system samples the performer, or the performer is playing electronic music);
- The creative focus on the system is on real-time interaction. This draws greater attention to the participatory interactive nature of the agent, rather than as a creative intelligence, which lessens the burden on the system designer to create a system capable of producing diverse and novel outputs.

For the present project, the genre we hope to target is electronic dance music (EDM). Because it is fully or predominantly electronic in its production, and thus avoids issues of expressive human performance [50], we feel that it offers great opportunities for MuMe practice; furthermore, metacreative research into this genre has already been undertaken [51, 52]. In 2013, the Algorave movement, based primarily around the musical practice of live coding [53], became well known. The 2013 MuMe Algorave (Sydney, 2013) showcased algorithmically composed electronic dance music as well as live coding performances. In this context, however, the presentation of algorithmi-

[3] for example, see http://www.metacreation.net/mumewe2013/index.php?pg=program for the program of a MuMe Weekend.

[4] although it may be far from *inclusive*; it is still a genre with norms and expectations.

cally composed pieces was problematic given the lack of a live performer, as well as, related to this, the awareness that there was no need for the music to be produced in realtime.

Reflecting on this event, it was agreed that it would be more engaging to present the work in realtime, and in order to do so in a meaningful way, agents needed to be placed in a context where they were responding to something. Responding to the audience was considered, as was responding to a data source, but this was deemed too gimmicky and too far removed from the immediate challenges of creating powerful, compelling musical metacreative systems. Thus it was agreed that performances should be collaborative, with various agents contributing different elements of a piece of music. This context therefore embodies the common creative musical challenge of getting elements to work together, reconceived as a collective metacreative task. Thus although the metaphor of a *jam* comes to mind in describing this interactive scenario, we prefer to imagine our agents acting more like the separate tracks in a carefully crafted musical composition.

We summarise the above motivating factors, adding a number of others below:

- Currently, collaborative music performance using agents is limited to human-computer scenarios. These present a certain subset of challenges, whereas computer-computer collaborative scenarios would avoid some of these whilst presenting others. Most existing live MuMe activity is limited to improvised genres such as jazz and blues, free improvisation, or scored genres such as classical music where the focus is on score following and expressive score interpretation. Endemically digital genres such as electronic dance music lend themselves well to musical metacreation, but currently offer no collaborative creation opportunities for MuMe practitioners in either live or non-live contexts;
- It allows us to build an infrastructure, which can be useful for commercial MuMe applications. Specifically, it provides a modular solution for the metacreative workstations of the future;
- It provides an easy way into MuMe methods and technologies, as musebots can take the form of the simplest generative units, whereas at present the creation of a MuMe agent is an unwieldy and poorly bounded task. Musebot ensembles can be educational, and could be used as the brief for an undergraduate course on generative music, or a workshop for one of the common computer music platforms, such as SuperCollider[5] or MaxMSP[6]. Musebots and musebot ensembles potentially could be experienced and interacted with (e.g., remixed) by non-programmers using end-user interfaces;
- It provides a platform for peer-review of systems and community evaluation of the resulting musical outputs, as well as stimulating sharing of code. We believe this will help clarify and provide a shared platform for reaching technical research goals in music AI, in a way that scientific communities such as the music information retrieval (MIR) community have managed through MIREX: the Music Information Retrieval Evaluation eXchange [54]. As MIREX illustrates, research progress can depend on articulating problems and building research infrastructure. An interesting additional direction is that musebots could be run in offline simulations to examine their dynamic properties, satisfying the proposal of Bown and Martin [55];
- It encourages and supports the creation of work in a publically distributed form that may be of immediate use as software tools for other artists;
- It outlines a new creative domain, which explores new music and music technology possibilities. The opportunities for remixing, mashing up, branching, coordinating, reappropriating and recontextualising musebots are all rich and open-ended areas of potential innovation;
- It defines a clear unit for software development. Musebots may be used as modular components in other contexts besides musebot ensembles. It modularises and distributes the task of automating composition, since the task of formalising aspects of musical composition in simple generative modules – such as a dubstep bassline generator, or a flamenco *cajón* player – is regularly and easily achieved. This also lowers the bar for entry.

[5] http://supercollider.sourceforge.net/
[6] https://cycling74.com/max7/

System Design

Technical and Aesthetic Design Considerations

How should an ensemble of musebots be coordinated? While ecosystemic or other bottom-up methods are conceivable [56], the use of a conductor agent that initiates performance and oversees general musical parameters such as tempo, time signature, and overall density is a more pragmatic solution: Eigenfeldt [57] offers one such implementation. The use of a conductor agent does not negate the potential for agents to communicate amongst themselves directly: the amount of attention paid to the conductor is open to the individual agents, although we do strictly require that agents respond to on/off and volume commands. We therefore nominate the “musebot conductor” (MC) as a central tool and guiding design principle for musebot ensembles.

Another immediate concern is how individual agents assume roles. Again, while the potential for self-organization is one option, issues of practicality suggest that pre-defined musical parts, or even collaboratively devised musebot ensembles, are more feasible. One possibility is that the conductor “build” an ensemble, by first assigning beat-generation to one musebot, then basslines to another, then harmonic aspects to a third, until a given set of musical roles are filled. Another possibility is that a prebuilt ensemble may be submitted in which these roles are already defined. More generally, these are decisions that each organiser of a musebot ensemble event might make differently. In order to cover all scenarios, we use a properties file, readable by both machines and humans that describe the capabilities and style of the individual musebot.

Clearly, potential designers of musebots will create software agents in unique ways. We assume that most designers will have some experience with metacreative systems, and an awareness of an important difference between human and virtual performers [58]; whereas human improvisers will expect other performers to be keen listeners that adapt accordingly, this is, in fact, extremely difficult to replicate in virtual agents. Instead, higher-level goals of emergence, conversation, and journey [59] are objectives that we may only hope to achieve incrementally [60]. For example, instead of designing a bass player that performs patterns for a specific drum pattern (a standard method of composing EDM), a more useful method would be to design a bass agent that can adapt to different drum beats (as provided by a separate beat agent), as well as a changing harmonic patterns provided by another agent. This is a good example of how we think musebot ensembles will drive innovation in the underlying principles of musical intelligence: for example, in establishing the relationship between a beat and a bassline, what simple pattern representations (as opposed to the musical *surface*) might best convey the musical sense.

The Musebot Agent Specification

An official musebot agent specification is maintained as a collaborative document, which can be commented on by anyone, and edited by the musebot team. An accompanying Github page maintains a repository of source samples and examples for different common languages and platforms.

A musebot ensemble consists of one MC and any number of musebots, running on the same machine or multiple machines over a local area network. As discussed above, the MC is responsible for high-level control of connected musebot agents in the network. The MC maintains control of the overall clock tempo of the ensemble performance and manages the temporal arrangement of agent performances (via volume mixing, on/off commands). The MC also assists connected agents inter-Musebot communication by continuously broadcasting a list of all connected agents to the network.

As a minimum requirement, a musebot agent must reveal itself to the MC upon connection to the network by providing a unique client ID and a periodic heartbeat, and respond to volume and shutdown commands received from the MC. All other communication is optional, although responding to the MC's timing messages is strongly recommended. Musebots may also broadcast any messages they want to the network, providing they maintain their unique name space allocated for inter-Musebot communication.

Our musebot specification states that a musebot should also "respond in some way to its environment", which may include any OSC messages as well as the audio stream that is provided (a cumulative stereo mix of all musebot agents actively performing). It should also not require any human intervention in its operation. Beyond these strict conformity requirements, the qualities that make a good musebot will emerge as the project continues.

An Example

For teaching, development and testing purposes, a number of elementary example musebots are provided along with the musebot agent specification, with source code and supporting libraries, developed using a range of the most popular creative computing environments (MaxMSP, PD, SuperCollider, Processing). Musebot creators can hack these examples to make their own agents, and we plan for additional creative computing environments to be catered for in time.

The example agents can be downloaded along with a draft MC application, and the ensemble can be run (in this initial case it is up to the user to manually turn on and off the agents). As the project progresses, this example will be joined by a growing population of musebot agents, and new versions of the MC. Instructions are provided for how a network of individual computers can be configured to run the ensemble in a distributed manner over a local network.

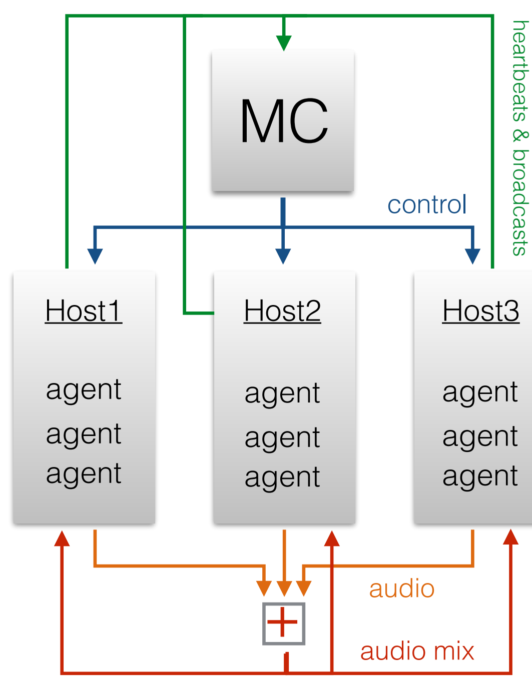

Figure 1: Interactions between agents and the MC.

The First Year of Musebot Ensembles

The musebot specification and call for contributions was published at the end of 2014 and a series of musebot ensemble events have been organised to occur throughout 2015. Although we would strive to provide existing musebot agents to designers in the future, for use in "rehearsing" their own agents, the premiere performance will most likely involve agents performing as an ensemble for the first time (if the ensemble has not been designed as a whole). In this case, it is envisaged that through consultation with the community in the open call process, a series of musebots will be designed that will compliment each other musically. This initial collection of musebots will provide the basis for future iterations of musebot ensembles, given the modular potential of this community-based approach. The premiere performance for the musebot platform will occur at the International Conference on Computational Creativity at Park City, Utah, in June 2015.

Managing Quality Control

An open call for works is all very well, but how should quality control be managed? The first thing to note is that in the spirit of openness and transparency, all submissions can be made public. Since the musebot conductor software is also publically available, anyone can draw on the entire public musebot repository to put together his or her own musebot ensemble.

An additional point to this is that, as with our more detailed example, we expect collaborations to form within the community, such that certain agents are designed with other specific agents in mind. Thus, the challenge of making sure agents actually work together is somewhat alleviated by the ongoing process of listening and improving how the agents respond to each other. We envisage that this emergent and collaborative process of design will engender a community-based approach to musical metacreation, based around shared code, open dialogue and modularised musical composition. In this context, over time the practice of musebot design could be expected to develop its own set of accepted methods and practices, with contributing designers building upon and learning from the creations of others in the community, much like other collaborative artistic endeavours.

Conclusion

This paper provides the thinking behind creating a community-driven musebot ensemble, and draws on the literature from a diverse range of computer music research to suggest a best-practice approach to creating the required infrastructure for this project. We refer readers to the ongoing progress of the project via the musebot specification[7] and software repository[8].

References

1. Gerhard Nierhaus, *Algorithmic composition: paradigms of automated music generation*, (Springer, 2009).
2. Palle Dahlstedt and Peter McBurney, "Musical agents: Toward Computer-Aided Music Composition Using Autonomous Software Agents," *Leonardo* (Vol. 39, no. 5, 2006): 469-470.
3. Robert Rowe, *Machine musicianship*, (MIT press, 2004).
4. Tim Blackwell and Michael Young, "Live algorithms," *Artificial Intelligence and Simulation of Behaviour Quarterly,* (Vol. 122, no. 7-9, 2005): 123.
5. Mitchell Whitelaw, *Metacreation: art and artificial life*, (Mit Press, 2004).
6. Somnuk Phon-Amnuaisuk and Geraint Wiggins, "The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system," *Proceedings of the AISB'99 Symposium on Musical Creativity*, (Edinburgh, 1999): 28–34.
7. Oliver Bown and Aengus Martin, "Autonomy in music- generating systems," *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, (Palo Alto, 2012).
8. William Hsu and Marc Sosnick, "Evaluating interactive music systems: An HCI approach," *Proceedings of New Interfaces for Musical Expression* (Pittsburgh, 2009).
9. Arne Eigenfeldt, Oliver Bown, Philippe Pasquier, and Aengus Martin, "Towards a taxonomy of musical metacreation: Reflections on the first musical metacreative weekend," *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, (Boston, 2013).
10. George Lewis, "Interacting with latter-day musical automata," *Contemporary Music Review,* (Vol. 18, no. 3, 1999): 99–112.
11. Robert Rowe, "Machine composing and listening with Cypher," *Computer Music Journal,* (Vol 16, no. 1, 1992): 43–63.
12. Matthew Yee-King, "An automated music improviser using a genetic algorithm driven synthesis engine," *Applications of Evolutionary Computing*, (Springer Berlin Heidelberg, 2007): 567–576.
13. Oliver Bown, Arne Eigenfeldt, Philippe Pasquier, Aengus Martin, and Benjamin Carey, "The Musical Metacreation Weekend: Challenges Arising from the Live Presentation of Musically Metacre-

[7] http://musicalmetacreation.org/musebots

[8] https://bitbucket.org/obown/musebot-developer-kit

ative Systems," *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, (Boston, 2013).

14. Jim Butterfield, Stan Ockers, and Eric Rehnke, *The First Book of KIM*, (Hayden Book Company, 1978).
15. John Bischoff, Rich Gold, and Jim Horton, "Music for an interactive network of microcomputers," *Computer Music Journal* (Vol. 11, no. 3, 1978): 24–29.
16. Scot Gresham-Lancaster, "The aesthetics and history of the hub: The effects of changing technology on network computer music," *Leonardo Music Journal* (Vol. 8, 1998): 39–44.
17. Gil Weinberg, "The aesthetics, history, and future challenges of interconnected music networks," *Proceedings of the International Computer Music Association Conference*, (Gothenburg, 2002): 349–356.
18. Alain Renaud, Alexander Carôt, and Pedro Rebelo, "Networked music performance: State of the art," *Proceedings of the AES 30th International Conference*, (Saariselkä, 2007): 16–22.
19. Matthew Wright, "Open Sound Control-A New Protocol for Communicating with Sound Synthesizers," *Proceedings of the 1997 International Computer Music Conference*, (Thessaloniki, 1997): 101–104.
20. Dan Trueman, "Why a laptop orchestra?" *Organised Sound* (Vol. 12, no. 2, 2007): 171–179.
21. Christopher Burns and Greg Surges, "NRCI: Software tools for laptop ensemble." *Proceedings of the International Computer Music Conference*, (2008), 344–347.
22. Ge Wang, Ananya Misra, and Perry Cook. "Building collaborative graphical interfaces in the audicle," *Proceedings of the 2006 conference on New interfaces for musical expression*, (IRCAM—Centre Pompidou, 2006), 49–52.
23. Juan-Pablo Cáceres and Chris Chafe, "JackTrip: Under the hood of an engine for network audio," *Journal of New Music Research,* (Vol. 39, no. 3, 2010): 183–187.
24. Josh Beggs and Dylan Thede. *Designing web audio*, (O'Reilly Media, Inc., 2001).
25. Lonce Wyse and Srikumar Subramanian, "The viability of the web browser as a computer music platform," *Computer Music Journal,* (Vol. 37, no. 4, 2013): 10–23.
26. Caleb Stuart, "The object of performance: Aural performativity in contemporary laptop music," *Contemporary Music Review,* (Vol. 22, no. 4, 2003): 59–65.
27. Kim Cascone, "Laptop music-counterfeiting aura in the age of infinite reproduction," *Parachute* (2002): 52–59.
28. Scott Smallwood, Dan Trueman, Perry Cook, and Ge Wang, "Composing for laptop orchestra," *Computer Music Journal,* (Vol. 32, no. 1, 2008): 9–25.
29. Ajay Kapur, Michael Darling, Dimitri Diakopoulos, Jim Murphy, Jordan Hochenbaum, Owen Vallis, and Curtis Bahn, "The machine orchestra: An ensemble of human laptop performers and robotic musical instruments," *Computer Music Journal,* (Vol. 35, no. 4, 2011): 49–63.
30. Jason Freeman and Akito Van Troyer, "Collaborative textual improvisation in a laptop ensemble," *Computer Music Journal,* (Vol. 35, no. 2, 2011): 8–21.
31. David Ogborn, "Composing for a Networked, Pulse-Based, Laptop Orchestra," *Organised Sound,* (Vol. 17, no. 01, 2012): 56–61.
32. EspGrid software, http://esp.mcmaster.ca/?page_id=1759
33. David Cope, *The algorithmic composer*, (Vol. 16, AR Editions, 2000).
34. Click Nilson, "Live coding practice," *Proceedings of the 7th international conference on New interfaces for musical expression*, (ACM 2007), 112–117.
35. Michael Wooldridge, *An Introduction to Multiagent Systems*, (John Wiley & Sons, 2009).
36. David Murray-Rust, Alan Smaill, and Michael Edwards, "MAMA: An Architecture for Interactive Musical Agents," *Frontiers in Artificial Intelligence and Applications,* (Vol. 141, 2006): 36.
37. Nick Collins, "Reinforcement learning for live musical agents," *Proceedings of the International Computer Music Conference,* (Belfast, 2008).
38. David Murray-Rust and Alan Smaill, "Towards a model of musical interaction and communication," *Artificial Intelligence,* (Vol. 175, no. 9, 2011): 1697–1721.
39. Aengus Martin, Craig Jin, and Oliver Bown, "A toolkit for designing interactive musical agents," *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, (ACM, 2011): 194–197.
40. Benjamin Carey, "Designing for Cumulative Interactivity: The _derivations System," *Proceedings of the 12th International Conference on New Interfaces for Musical Expression* (Ann Arbor, 2012).
41. Oliver Bown, Petra Gemeinboeck, and Rob Saunders, "The Machine as Autonomous Performer," *Interactive Experience in the Digital Age*, (Springer International Publishing, 2014): 75–90.
42. Benjamin Houge and Jutta Friedrichs. "Food Opera: A New Genre for Audio-gustatory Expression," *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2013.

43. StyleMachine Lite by Metacreative Technologies, http://metacreativetech.com/products/stylemachine-lite/
44. Oliver Bown and Sam Britton, "An Album in 1,000 Variations: Notes on the Composition and Distribution of a Parametric Musical Work," *Leonardo,* (Vol. 47, no. 5, 2014), 437–441.
45. Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov, "Omax brothers: a dynamic topology of agents for improvization learning," *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, (ACM, 2006): 125–132.
46. Alice Eldridge, "Collaborating with the behaving machine: simple adaptive dynamical systems for generative and interactive music," *PhD diss.* (University of Sussex, 2007).
47. Oliver Bown, Petra Gemeinboeck, and Rob Saunders, "The Machine as Autonomous Performer," Interactive Experience in the Digital Age, (2014).
48. Tim Blackwell, and Michael Young. "Self-organised music." *Organised Sound*9, no. 02 (2004): 123-136.
49. Tim Blackwell, Oliver Bown, and Michael Young, "Live algorithms: towards autonomous computer improvisers," Computers and Creativity, (Springer Berlin Heidelberg, 2012): 147–174.
50. Alexis Kirke and Eduardo Miranda. *Guide to computing for expressive music performance*, (Springer, 2013).
51. Dimitri Diakopoulos, Owen Vallis, Jordan Hochenbaum, Jim Murphy, and Ajay Kapur, "21st Century Electronica: MIR Techniques for Classification and Performance," *International Society for Music Information Retrieval Conference*, (Kobe, 2009), 465–469.
52. Arne Eigenfeldt and Philippe Pasquier, "Evolving structures for electronic dance music," Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference," (ACM, 2013): 319–326.
53. Nick Collins and Alex McLean, "Algorave: A survey of the history, aesthetics and technology of live performance of algorithmic electronic dance music," *Proceedings of the International Conference on New Interfaces for Musical Expression*, (London, 2014).
54. Stephen Downie, "The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research," *Acoustical Science and Technology* (Vol. 29, no.4, 2008): 247–255.
55. Oliver Bown and Aengus Martin, "Autonomy in music- generating systems," 2012.
56. Jon McCormack and Oliver Bown, "Life's what you make: Niche construction and evolutionary art," *Applications of Evolutionary Computing*, (Springer Berlin Heidelberg, 2009): 528–537.
57. Arne Eigenfeldt, "Emergent rhythms through multi-agency in Max/MSP," *Computer Music Modeling and Retrieval. Sense of Sounds*, (Springer Berlin Heidelberg, 2008): 368–379.
58. David Borgo, *Sync or swarm: Improvising music in a complex age*. (Bloomsbury Publishing, 2005).
59. David Borgo, "Emergent Qualities of Collectively Improvised Performance," *Pacific Review of Ethnomusicology* (Vol. 8, no. 1, 1996-1997): 23–40.
60. Tim Blackwell, Oliver Bown, and Michael Young, "Live algorithms: towards autonomous computer improvisers," *Computers and Creativity*, (Springer Berlin Heidelberg, 2012): 147–174.

Author Biographies

Ollie Bown is a researcher, programmer and electronic music maker. His research is concerned with creative computing (the tools and programming languages that enable the production of creative outputs), computational creativity (the modeling of creative processes using software) and the social role and evolutionary origins of music.
He creates and performs music as one half of the duo Icarus, and performs regularly as a laptop improviser in electronic and electroacoustic ensembles. Icarus' 2012 album Fake Fish Distribution was released in 1000 unique digital variations, presenting a radical conception of ownership and uniqueness in digital media artefacts. In his research role he was the local co-chair of the 2013 International Conference on Computational Creativity and of the Musical Metacreation Workshop and events series.

Benjamin Carey is a Sydney-based saxophonist, composer and technologist with interests in contemporary classical, interactive, improvised and electro-acoustic music. His recent research and practice incorporates equal parts improvisation, composition and the development of musical software systems. His work has been featured at numerous international festivals and conferences including the dBâle festival of electronic music (Basel), IRCAM Live @ La Gaité Lyrique (Paris), Vivid Live (Sydney), the Australasian Computer Music Conference (Brisbane/Auckland), the Conference on New Interfaces for Musical Expression (Ann Arbor/London), and the International Computer Music Conference (Perth).

Arne Eigenfeldt is a composer of live electroacoustic music, and a researcher into intelligent generative music systems. His music has been performed around the world, and his collaborations range from Persian Tar masters to contemporary dance companies to musical robots. He is a co-director of Metacreation Lab at Simon Fraser University, where he is a Professor of Music and Technology.