

# Musebots at One Year: A Review

Arne Eigenfeldt

School for the Contemporary Arts, Simon Fraser University, Vancouver, Canada  
arne\_e@sfu.ca

## Abstract

A musebot is defined as a piece of software that autonomously creates music collaboratively with other musebots. The musebot project is concerned with putting together musebot ensembles, consisting of community-created musebots, and setting them up as ongoing autonomous musical installations. The specification was released early in 2015, and several developers have contributed musebots to ensembles that have been presented in the USA, Canada, and Italy. To date, there are over sixty publically available musebots. Furthermore, the author has used the musebot protocol in several personal MuMe projects, as they have provided a flexible method for generative systems in performance and installation. This paper will review the past year, and how musebots have been used in both their original community-oriented installations, as well as the author's works.

## Introduction

Musebots are pieces of software that autonomously create music, collaboratively with other musebots. A defining goal of the musebot project [Bown *et al.* 2015] is to establish a creative platform for experimenting with musical autonomy, open to people developing cutting-edge music intelligence, or simply exploring the creative potential of generative processes in music. Not simply a robot jam, but individual virtual instrumentalists coming together, like a band, to autonomously create music.

The second aim of the Musebot project is to establish a playful and experimental platform for research, education and making, that will stimulate interest and advance innovation in musical metacreation (MuMe). Above all, the musebot project is a collaborative, creative experiment: we have invited others in the generative music community to join us in making autonomous software agents that work together to make original music; to date, seven developers have contributed to over five dozen musebots, written in MaxMSP, Java, PD, Extempore, and Max for Live.

A larger and more long-term goal for the project has been a sharing of ideas, as well as code. There has been a lot of research in MuMe systems, and the results are impressive. But a lot of the creative work is in idiosyncratic, ad hoc standalone systems, and their results can be opaque. It is difficult for artistic researchers to share their ideas or their code, or work out ways that their systems might be incorporated into other's creative workflows. Musebots, by contrast, are small modular units that are designed to be shared and studied by others. By making collaboration central, the musebot project forces us to be transparent in how our systems work [Eigenfeldt *et al.* 2015].

The initial deployment of musebots was within an electronic dance music (EDM) aesthetic, primarily because it is fully or predominantly electronic in its production, and the installations, described below, have maintained this aesthetic. However, the author has used musebots in other MuMe creative research, including as generators for robotic instruments in performance and installations (*MachineSongs*), as part of a generative music-sound-video system (*Seasons*), and as an exploration of Moment-form (*Moments*).

## Description

Each musebot agent corresponds roughly to a single "instrumental part" in a piece of music, such as a bassline, a drumbeat, or synth part; however, the agents should be smart, in that they should be able to react to their environment, and communicate their current states. The musebot protocol is, at its heart, a method of communicating states and intentions, sending networked messages established through a collaborative document via OSC [Wright 1997]. A Conductor serves as a running time generator, as well as a hub through which all messages pass. The Conductor also launches individual musebots via curated *ensembles*. The protocol is more fully described elsewhere [Bown *et al.* 2015; Eigenfeldt *et al.* 2015]. Currently, there are over sixty musebots in the repository: over half of which have

been created by the author; the remainder by his graduate students and other developers.

## Creative Use

Although the author has used multi-agents in a variety of contexts [Eigenfeldt 2007, Eigenfeldt 2010; Eigenfeldt and Pasquier 2011], their use was always idiosyncratic to the composition. What has proven to be particularly appealing to the author has been the flexibility of musebots, as well as their open-endedness. Musebots designed for a particular work can be easily adapted to other contexts by altering the type of messages communicated.

### Musebot Chill-out Sessions

The premiere of musebots occurred in July 2015 as an installation at the International Conference on Computational Creativity (ICCC) in Park City, and was followed in August 2015 at the International Symposium of Electronic Art (ISEA) in Vancouver. It was presented at the Generative Art Conference in Venice in December 2015, and will be installed at the New Interfaces for Digital Expression (NIME) conference in Brisbane in July 2016. The first musebot ensembles are more fully described elsewhere [Eigenfeldt *et al.* 2015], along with issues and questions raised.

The Chill-out Sessions – so named due to an initial desire to provide musebots as an alternative listening space to the dance rhythms of Algoraves – have consisted of fifty curated ensembles of available musebots. Each ensemble consists of musebots providing the typical instrumental parts within EDM: beats, bass, harmony, and various synth parts, as well as a tempo and duration. Some ensembles are more ambient – slower tempi with musebots that produce less aggressive beats and more continuous sustained synthesiser textures – while others are combinations of only beat-generating musebots.

When an ensemble is loaded, each musebot, which must be a standalone application, is successively launched and immediately begins sending a heartbeat. The Conductor compares the cumulative heartbeats with the ensemble list: if they match, the Conductor initiates its timecode messages; if not (for example, if a musebot fails to launch), the Conductor quits all musebots, and loads the next ensemble. All musebots can begin playing as soon as the timecode is received; certain musebots may depend upon external messages – for example, a current harmony (called *notepool*) produced by harmony-generating musebots – and wait for specific messages; other musebots may generate an overall form for themselves, and not begin playing until their form dictates they should do so.

One aspect of the musebot specification suggests that musebots include an `info.txt` file that lists messages it

transmits and to which it responds; the goal being that messages may develop dependent upon musical requirements. One such message that the author uses is density: the relative number of onsets currently playing. Several beat-generating musebots have dynamic density levels, and when combined with other musebots that react to the density message, produce music that seems intentional.

In addition to hand-curated ensembles, a curation algorithm was coded. This algorithm is based purely upon musebot type; for example, it avoids pairing multiple bass-generating musebots. We foresee that this algorithm could become much more powerful by using information found in the `info.txt` file once developers explore a greater variety of messages.

Durations of generated compositions for each ensemble have been limited to five minutes; some ensembles that make use of a delayed musebot launch feature have enjoyed a longer duration. Example video of Musebot Chill-out sessions can be found here: <http://tinyurl.com/h5b453g>

### MachineSongs - Live Performance

Although musebots were originally designed for machine-to-machine interaction, live human control is certainly possible. A particular performance of the author required a live guitarist to interact with a Metacreative system that controlled three musical robots; incorporating musebots was a logical choice for intelligent control. Musebots, according to the musebot specification, should produce their own sound so as to allow for their use on any system; however, since *MachineSongs* would only be performed under the composer's control, the musebot output was sent as MIDI information to the robotic instruments. Existing musebots were reconfigured, with beat-generating musebots controlling a percussion robot, and melodic and bass musebots controlling a marimba robot and a Disklavier.

Instead of relying upon ensembles to launch musebots autonomously via the Conductor, the author initiated the musebots individually in response to the live guitarist. Furthermore, a musebot controller – an extension to the Conductor – was created that controlled musebot dynamics (by sending volume messages) as well as unique valance (complexity) and arousal (density) messages. All musebots used were modified to respond to these two messages.

A recording of *MachineSongs*, as performed at ISEA 2015, can be found here: <https://youtu.be/0GzUV8afZiE>

### MachineSongs - Installation

The musebots designed for the live *MachineSongs* performance were combined with the original Conductor and its ensembles, and performed as an ongoing installation for one month. The work's program notes are as follows: "*MachineSongs* uses a Disklavier, the ModulatroN, a 35-mallet marimba-playing robot, and the Notomoton, an 18-

armed percussion-playing robot, both designed and built by Ajay Kapur's team at CalArts, all under the control of autonomous musebots. Using a machine analysis of a wide selection of music – including Mozart, Miles Davis, Pat Metheny, and 1940's swing music – the software generates a short composition using what it has learned from the corpus, in terms of melody, harmony, and rhythm. However, the unusual orchestration and performance results in a somewhat tongue-in-cheek realisation."

An example recording of the *MachineSongs* installation can be found here: <https://youtu.be/DtMqeorhEgA>

### **Seasons - A Multimedia Video and Sound Generative System**

The author is a member of a collaborative team in which generative music, sound, and video systems are combined to produce an audio-visual experience that models and depicts our natural environment across the span of a year: *Seasons*. The video sequencing engine selects several related videos based upon hand-coded metatags, and this information is sent to the soundscape and music systems. The original music generation system by the author was replaced with musebots, due to their flexibility and potential to adapt to information received from the video engine. Each new season triggers a new musebot ensemble, allowing for a wide variety of music generation. Furthermore, since musebots had been developed to react to valence and arousal, these parameters were used within the entire system to supplement the metatags sent by the video sequencer, resulting in a better affective relationship between video and music [Eigenfeldt *et al.* 2015b].

Example video from *Seasons* is available here: <https://vimeo.com/136361163>

### **Moments - An Exploration of Moment-form**

An open problem in musical metacreation is the generation of musical form [Eigenfeldt 2014]. The author believes that musebots are one potential avenue in which to pursue the generation of such high-level musical structures, as proposed in a recent paper with like-minded creative researchers [Eigenfeldt *et al.* 2016]. While that paper outlines many of the complexities of generating form, it reminded the author of Stockhausen's innovative contribution of Moment-form [Stockhausen 1963]: music based upon stasis, rather than motion and goals, in which Moments are offset by discontinuities between successive musical features. This prompted the author to create a series of musebots that generate much more consistent gestures than usual, coordinated by a *parameterBot* that provides a hierarchical means of controlling disparate features, such as voice density, activity level, complexity, and consistency, to name a few. Given a Moment's conditions, as requested by the *parameterBot*, individual musebots compare

their own potential to the current constraints, and decide whether to participate. *Moments* continues to be developed: a version will be premiered at *NIME* 2016.

## **Conclusions and Future Work**

Musebots have proven to be a flexible method for MuMe creation in a variety of situations, as demonstrated in their first year presented in this paper; however, their deployment as adaptive musical agents still needs to be pursued further and in greater depth. Because the author has been the main developer of musebots to date, their potential beyond the requirements of the author has been limited.

Several important interactions did occur between developers in the first year, including involvement in a ProcJam (see <https://bencarey.itch.io/musebots-for-procjam-2015>), which resulted in the formulation of a new intention message. It is hoped that more MuMe creators will adopt their use, and more such interactions will occur, resulting in true collaborative creation, a sharing of code, and more importantly, musical ideas within MuMe.

## **References**

- Bown, O., Carey, B., Eigenfeldt, A. 2015. *Manifesto for a Musebot Ensemble: A platform for live interactive performance between multiple autonomous musical agents*. In Proceedings of the International Symposium of Electronic Art, Vancouver.
- Eigenfeldt, A. 2007. *Emergent rhythms through multi-agency in Max/MSP*. In Computer Music Modeling and Retrieval. Sense of Sounds. Springer Berlin Heidelberg.
- Eigenfeldt, A. 2010. *Coming together: Composition by negotiation*. In Proceedings of the ACM International Conference on Multimedia, Florence.
- Eigenfeldt, A. 2014. *Generating Structure – Towards Large-scale Formal Generation*. In Proceedings of the International Conference on Musical Metacreation, Raleigh.
- Eigenfeldt, A., and Pasquier, P. 2011. *Negotiated Content: Generative Soundscape Composition by Autonomous Musical Agents in Coming Together: Freesound*. In Proceedings of the International Conference on Computational Creativity (ICCC), Mexico City.
- Eigenfeldt, A., Bown, O., and Carey, B. 2015a. *Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble*. In Proceedings of the ICC, Park City.
- Eigenfeldt, A., Bizzocchi, J., Thorogood, M., and Bizzocchi, J. 2015b. *Applying Valence and Arousal Values to a Unified Video, Music, and Sound Generative Multimedia Work*. In Generative Art Conference, Venice.
- Eigenfeldt, A., Bown, O., Brown, A., Gifford, T. 2016. *Flexible Generation of Musical Form: Beyond Mere Generation*. In Proceedings of the ICC, Paris.
- Stockhausen, K. 1963. *Momentform*. In Texte zur Musik 1.
- Wright, M. 1997. *Open Sound Control-A New Protocol for Communicating with Sound Synthesizers*. In Proceedings of the International Computer Music Conference, Thessaloniki.