Co-Creativity with a Musical AI Agent in Live Coding

ANONYMOUS AUTHOR(S)*

Algorithms (of some sort or another) have been present in music for centuries. The practice of live coding, has adopted them in live musical performance settings as a new paradigm of highly-expressive music making. Here, the notions of creativity between humans and computers, and the interaction between the two parties, are considered for live coding systems. Specifically, this work focuses on how computationally creative systems based in live-coding should incorporate emotional meaning into collaborative musical actions by looking to models of affective response.

CCS Concepts: • **Human-centered computing** → *Sound-based input / output*; Interaction design theory, concepts and paradigms.

Additional Key Words and Phrases: Musical Artificial Intelligence , Computational Creativity , Live Coding, Co-creation

ACM Reference Format:

Anonymous Author(s). 2022. Co-Creativity with a Musical AI Agent in Live Coding. In . ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Co-creation strategies for human-machine collaboration have been explored in various creative disciplines. Recent developments in music technology and artificial intelligence have made these creative interactions applicable to the domain of computer music, meaning it is now possible to interface with algorithms as creative partners. The application of this research is incorporated within the context of a specific field of algorithmic composition known as live coding. As music is inherently coupled with affective response, it is crucial for any artificial musical intelligence system to consider how to incorporate emotional meaning into collaborative musical actions. This work will look at bestowing machine musicians within interactive live coding systems the ability to create affective musical collaborations and examine new ways of interfacing with musical algorithms.

This paper discusses the development an agent in the field of musical artificial intelligence in an attempt to categorise behaviours in co-creativity between a machine agent and human agent improvising music. Three key components are proposed for successful collaboration with a machine musician: the ability of the musician to create music from a model of human affect, the role of machines to develop (artificial) aesthetics, and the methodology by which these facilities of collaboration can be made possible and effective for the human-musician.

2 LIVE CODING

Live coding is a term often used to refer to the act of creating art by writing computer code, usually in front of an audience [2]. In live coding, computer language is the primary medium for notation and describing the rules with which to synthesise artworks, and in this case we consider the case where the output is musical. Live coding is a subset of the wider field of *algorithmic composition* - where music is generated through finite sequences of well-defined, computer-implementable instructions. In live coding the performance comes from the *process* of software development,

Manuscript submitted to ACM

rather than its outcome. The work is not generated by a finished program, but through its journey of development from nothing to a complex algorithm, generating continuously changing musical or visual form along the way. With live coding, the human act of programming provides the creative impetus [9].

Although natural language and programming languages are ontologically distinct, programming languages provide a way of interfacing with computers and music technology in a *human* way. Language has historically been used as an artistic medium due to its capacity for emotional expression through uses of narrative and imagery. Music and language have also historically been symbiotic, with the contemporary artistic practice of musical live coding attempting to furthering this symbiosis. In live coding, computer programming languages are the method by which to notate musical ideas. Because of its linkages to language, live coding is thus the ideal breeding ground for novel works in collaborative and co-creative systems with machine musicians through harnessing the power and affordances of language.

```

d4
$ every 4 (0.25 <~)
$ every' 8 6 (brak)
$ someCyclesBy 0.35 (randDelay)
$ almostNever ((# speed "-1").(#acc "-0.8"))
$ stack [
  sound "~ ~ ~ jhit:1" # speed 1.5 # gain 0.8,
  sound "sfe:4(9,16)?" # speed 1.5 #accelerate "0.2 0.8 0.01 -1"
   #gain 0.8,
  sometimes (fast "0.5" ) $ gain "1.05*16?"
  # sound (choose  ["revkit", "007kit1"])
  # n (irand 16) # speed "[0.75 0.5]/16" #accelerate "0.2 0.8 0.01 -1"
]
#nudge 0.21
|*| gain 0.99

```

Fig. 1. An example of code in the TidalCycles live-coding language for producing musical pattern. As an extension of the Haskell language, this code is purely functional and allows high-level manipulation of pattern and types can be inferred by the compiler, which leads to less interruptions to the "liveness" of the system.

3 BUILDING AN AFFECTIVE MODEL FOR A COMPUTATIONALLY CREATIVE SYSTEM

Boden addresses machine creativity in artificial intelligence, presupposing it as an inherently human trait and defining it as "the ability to generate novel, and valuable, ideas" [1]. However, for artificial musical intelligence to be successful in its aim of creation of valuable music, more focus is needed on the subjective human response rather than generation based purely on musical surface form. Approaching the task of music generation from a purely computational standpoint detaches it from its essence of inherent emotional expression. This is clear in automatic language generation, given most linguistic systems examine the narrative and its intended message rather than merely generating based on syntactical input alone, and so the same considerations are required for musical systems [14].

Music is intrinsically tied to its elicited affective response. Many studies link its importance with memory and perception [4], indeed it is perhaps defined by these factors [13]. Thus, this research looks at how to incorporate aspects of affective response modelling into the processes by which a system is computationally creative.

Various methodologies can be used to implement the modelling of affective response. The two main modes for emotional representation are categorical (using discrete labels) and dimensional (affective phenomenon as a set of coordinates in a low-dimensional space). Many works in the field however tend towards using what is known as the circumplex model with an x-axis of valence (how positive/negative the affective state is) and y-axis of arousal (how low/high energy the affective state is). For example, in the METACOMPOSE system [11], compositions are created in real-time which express mood-states based on this model for emotional affect. Similarly, [8] use a rule-based approach in their affective algorithmic system, CMERS, a Computational Music Emotion Rule System. It maps emotions from the plane of valence and arousal to the structural and performative features of a generated composition. This is paired with a visual interface which allows the user to select their desired levels of valence and arousal for the music production.

In this research, a dimensional approach based on the valence-arousal co-ordinate space is used. This is the basis by which an affective model is created for the generation of new musical compositions. This takes the circumplex model parameterised by valence and arousal co-ordinates, $v, a \in [-1, 1]$. The proposed approach looks to existing literature on music and emotion [6] and how this can be incorporated to build accurate models of affective response within the musical elements of the composition.

The relationships proposed based on the literature build a mapping from intended target valence-arousal parameters onto the musical structural aspects of tempo, loudness, rhythmic density, pitch register, pitch contour and modality. Some literature suggests the use of three-dimensional models for additional clarification (with added dimensions such as tension or dominance) [10] however these are omitted from the current research at this stage, due to the constraints of the online listening test having to rate on any interface with more than 2-dimensions requiring specialised equipment.

Validation of these models in representing affective correlates occurred through online listening tests that took place in November - December 2020. From this, data was obtained on how well the intended mapping of affective co-ordinates matched with the perceived evaluation of the generated patterns' audio output.

The data was grouped into two types of stimuli: generated melodic patterns and generated rhythmic patterns. There were nine groupings of stimuli, for conditions generated under groupings of (low/mid/high) valence/arousal. In other words, this system was confirmed to correctly communicate emotional content if:

(1) Listeners perceived (low/mid/high) valence when the system was configured to generate music with (low/mid/high) valence values.
(2) Listeners perceived (low/mid/high) arousal when the system was configured to generate music with (low/mid/high) arousal values.

Preliminary analysis on this data suggested there was significant and notable correlations between the intended target parameters of valence and arousal and the observed parameters. The observed correlations coefficients used the Spearman's rank method. This was taken using the whole dataset correlated with the intended parameter values. The values computed were $R = 0.88$ and p−value of $p = 0.0031$. for the valence values of the various melodic stimuli, $R = 0.4$, $p = 2.8 \times e^{-16}$ for the arousal values of the various melodic stimuli, $R = 0.24$, $p = 4.9 \times e^{-5}$ for the valence values of the rhythmic stimuli and $R = 0.75$, $p = 2.2 \times e^{-16}$ for the arousal values of the rhythmic stimuli.

Given there is significant evidence to suggest that the mappings between valence and arousal and the structural parameters of the music generated, this will then be incorporated as part of the algorithmic composition strategies employed by the creative agents to allow control over these structural aspects of the music. The user of such a system could then request patterns from a creative agent partner of a specific affective content, much how musicians improvising together might work together to produce music of specific affective content.

4 BUILDING CO-CREATIVE SYSTEMS

Building systems for co-creation between a machine musician and a human musician is one of the overall aims of this research. Given the prolific climate for artificial intelligence and live-coding's grounding in human-computer interaction, it is unsurprising that the challenge of co-creation with machine musicians has already been attempted. Notable examples that generate Tidal code include an autonomous performer, *Cibo*, which tokenises Tidal code and uses a sequence-to-sequence neural network trained on a corpus of Tidal performances to generate novel patterns [12] or using a defined-grammar and evolutionary algorithms to evolve patterns, using the collaborative live-coding platform *Extramuros* [3].

Some steps have been made to complete a computationally creative system based in the live coding language TidalCycles, which are outlined in [15]. The TidalCycles language (an extension of the strongly typed functional programming language Haskell) is used for the generation of new musical patterns. This is integrated by allowing automatic suggestion of the agent's patterns to a live coder. We aim for this to be a co-creative system, using machine agents to explore not-yet conceptualised code sequences and support coders in asking new questions.

Figure 2 shows the overall structure of how the creative system creates new patterns of code. The system uses a random walk process, however rather than traditional methods where the musical parameters are the states of the Markov chain, this uses each Tidal "function" (i.e. the low-level words) to produce the new sequences. This is supported by Haskell's strict type-system that means that every sequence produced by the agent should be hypothetically executable. The process is as follows. Firstly, a small corpus of code was collected from the community. This corpus was then converted so that each function is treated as a single token. From the tokenisation, a n-gram model was built, that sorted the corresponding frequencies of n-length tokens. This was turned into a lookup function, that would give the frequency of any possible next-in-sequence tokens. This was passed to the type-checker, an inherent structure which provided all the possible functions that could follow in the sequence. The random walk was then performed through the possible search-spaces. The code was also filtered to remove any redundant output. Finally, the code output was passed back to the user interface so that the user can select or reject.

Currently, the affective model is not implemented in the generation process, so the produced code patterns are not based on the affective modelling. The final part of this research looks to combine the two aspects to achieve the aims outlined through this paper. Research into reinforcement learning are beginning to produce some crucial results for this implementation.

5 EVALUATION OF CREATIVE SYSTEMS

For music generation systems, it is often useful to evaluate the music produced, and can sometimes even be useful within the generation process. The musical output could either be evaluated by either human listeners or machine-learning models. The evaluation method is dependent on the goal of the system, such as its similarity to a particular style or corpus, or to sound *believably human.*

Current research in interactive music and computationally creative systems only sometimes employ formal evaluation methods and many systems are not described in sufficient detail for their re-implementation. As well as the design of a system for interaction with affective autonomous agents, this work intends to apply evaluation metrics to any musical

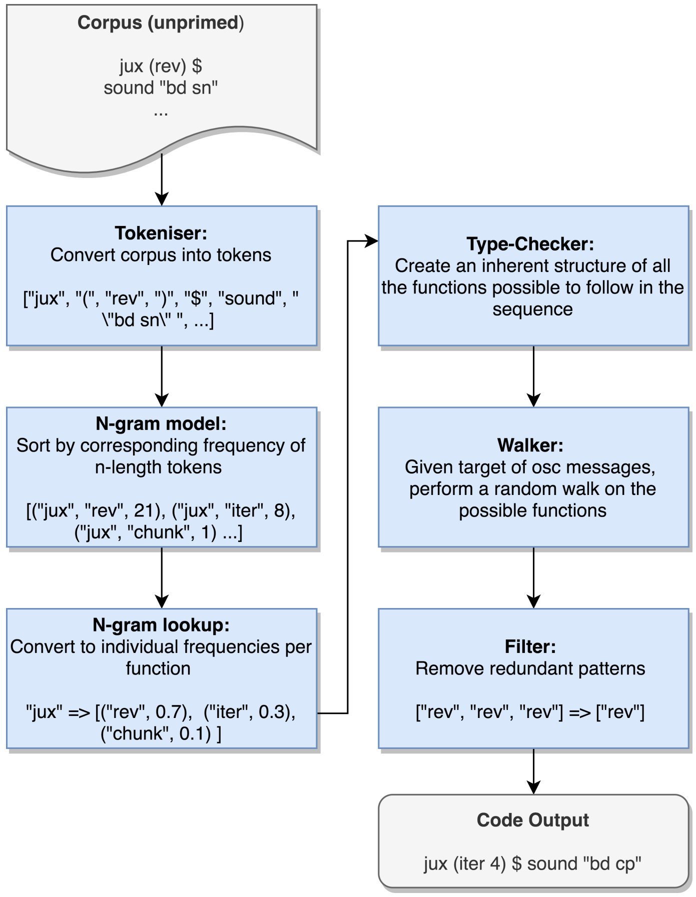

Fig. 2. Overall structure of the live-coding autonomous agent, taking an input corpus and using a random walk process on Haskell's strict type-system to produce new sequences of musical pattern.

output of the system. Some existing frameworks exist that aim to provide universal evaluation of computationally creative systems. Jourdanous proposes a Standardised Procedure for Evaluating Creative Systems (SPECs) [5], its approach based around a set of 14 'components of creativity' that evaluators should consider. Kantosalo also identifies this disparity between the production and evaluation of creative systems and proposes hybrid approaches from field of user-experience design and computational creativity research [7]. This work will aim to contextualise the existing research on evaluation of computationally creative systems within the framework of live coding. However, as the research on this specific domain is currently sparse, perhaps a significant contribution of this project is the application and adaption of existing methodologies to create new critical evaluation models of both musical output that is produced and the interaction between human and machine agents.

6 ACKNOWLEDGMENTS

This work was supported by EPSRC and AHRC under the EP/L01632X/1 (Centre for Doctoral Training in Media and Arts Technology) grant.

REFERENCES

[1] Margaret A Boden. 2004. *The creative mind: Myths and mechanisms.* Routledge.
[2] Nick Collins, Alex McLean, Julian Rohrhuber, and Adrian Ward. 2003. Live coding in laptop performance. *Organised sound* 8, 3 (2003), 321–330.
[3] Simon Hickinbotham and Susan Stepney. 2016. Augmenting live coding with evolved patterns. In *International Conference on Computational Intelligence in Music, Sound, Art and Design.* Springer, 31–46.
[4] Lutz Jäncke. 2008. Music, memory and emotion. *Journal of biology* 7, 6 (2008), 1–5.
[5] Anna Jordanous. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4, 3 (2012), 246–279.
[6] Patrik N Juslin and John A Sloboda. 2001. *Music and emotion: Theory and research.* Oxford University Press.
[7] Anna Kantosalo and Sirpa Riihiaho. 2019. Experience evaluations for human–computer co-creative processes–planning and conducting an evaluation in practice. *Connection Science* 31, 1 (2019), 60–81.
[8] Steven R Livingstone, Ralf Muhlberger, Andrew R Brown, and William F Thompson. 2010. Changing musical emotion: A computational rule system for modifying score and performance. *Computer Music Journal* 34, 1 (2010), 41–64.
[9] Christopher Alex McLean. 2011. *Artist-programmers and programming languages for the arts.* Ph. D. Dissertation. Goldsmiths, University of London.
[10] Klaus R Scherer, Tom Johnstone, and Gundrun Klasmeyer. 2003. Vocal expression of emotion. *Handbook of affective sciences* (2003), 433–456.
[11] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. 2017. Affective evolutionary music composition with MetaCompose. *Genetic Programming and Evolvable Machines* 18, 4 (2017), 433–465.
[12] Jeremy Stewart and Shawn Lawson. 2019. Cibo: An Autonomous TidalCyles Performer. In *International Conference on Live Coding.*
[13] Geraint A Wiggins. 2009. Semantic gap?? Schemantic schmap!! Methodological considerations in the scientific study of music. In *2009 11th IEEE International Symposium on Multimedia.* IEEE, 477–482.
[14] Geraint A Wiggins. 2018. To play with feeling? The opportunity of aesthetics in computational musical creativity. CEUR Workshop Proceedings.
[15] Elizabeth Wilson, Shawn Lawson, McLean Alex, and Jeremy Stewart. 2021. Autonomous Creation of Musical Pattern from Types and Models in Live Coding. In *9th Conference on Computation, Communication, Aesthetics & X.*