

On the Integration of Machine Agents into Live Coding

ELIZABETH WILSON¹ , GYÖRGY FAZEKAS¹ and GERAINT WIGGINS^{1,2}

¹Centre for Digital Music, Queen Mary University, London, UK

²AI Lab, Vrije Universiteit Brussel, Brussels, Belgium

Emails: elizabeth.wilson@qmul.ac.uk; george.fazekas@qmul.ac.uk; geraint.wiggins@qmul.ac.uk

Co-creation strategies for human–machine collaboration have recently been explored in various creative disciplines and more opportunities for human–machine collaborations are materialising. In this article, we outline how to augment musical live coding by considering how human live coders can effectively collaborate with a machine agent imbued with the ability to produce its own patterns of executable code. Using machine agents allows live coders to explore not-yet conceptualised patterns of code and supports them in asking new questions. We argue that to move away from scenarios where machine agents are used in a merely generative way, or only as creative impetus for the human, and towards a more collaborative relationship with the machine agent, consideration is needed for system designers around the aspects of reflection, aesthetics and evaluation. Furthermore, owing to live coding’s close relationship with exposing processes, using agents in such a way can be a useful manner to explore how to make artificial intelligence processes more open and explainable to an audience. Finally, some speculative futures of co-creative and artificially intelligent systems and what opportunities they might afford the live coder are discussed.

1. INTRODUCTION

In looking to build musical performances with machine agents in live coding, we discuss how to integrate elements into live coding practice so that the performance is co-creative. The term ‘machine agent’ is used where the agent has the capacity to produce its own sequences of executable patterns of musical code. To do this, knowledge from the field of computational creativity is translated and applied to live coding practices. In particular, this work is framed around Boden’s seminal works in the field of computational creativity (Boden 2004). The notion of a conceptual space and the exploration of it by creative agents are central to Boden’s model of creativity. The conceptual space is a collection of artefacts, or *concepts* in Boden’s terminology, viewed as appropriate representations of whatever is being made. The artefacts in our conceptual space are the possible patterns of code.

This article will first propose some epistemological framing for the use of musical agents in live coding. It will then look at shaping the co-creative process from a high-tech puppetry relationship to a more unilateral,

equal relationship where both the human and the machine live coders have creative agency. Co-creation is an intellectual and performative process and should happen with some intentionality on the part of the agent. We argue that to ‘collaborate with’ rather than simply ‘using’ a creative system in live coding, the systems built should contain (at least) these three elements: capacity for reflection, aesthetic consideration and evaluation.

2. EPISTEMOLOGICAL FRAMING FOR MUSICAL AGENTS IN LIVE CODING

Live coding, by its definition, generally entails some notion of ‘on-the-fly’ improvisation, or to a lesser extent, uncertainty. However definitions of liveness can vary in their semantics. One of its given definitions describes liveness as ‘of or involving a presentation (such as a play or concert) in which both the performers and an audience are physically present’ (Merriam-Webster 2022), while another is given as ‘broadcast directly at the time of production’ (ibid.). The former definition indicates the importance of an audience that tends to be a key feature of live coding performances, while the latter focuses more on the interdependence of timing and creation. Perhaps the latter definition being prefixed by the clause ‘broadcasting’ indirectly suggests an audience. Live coding performances these days are increasingly often broadcast, where the live coder could be performing in isolation in one location and the audience are distributed in other remote locations. These remain valid expressions of live coding without the physical presence attribute of the first definition.

Then does liveness predicate an audience? Can ‘live’ still occur when there’s no one around to hear it? If a live coder makes music in a solitary, isolated anechoic chamber and there is no one around to hear it, is it still live coded music? Tanimoto (2013) provides a historical perspective of liveness in programming. Tanimoto’s definition of liveness as ‘concern[ing] the timeliness of execution feedback’ (ibid.: 31) tends towards our second, audience-independent definition of liveness.

What does all this mean for the ecological validity of live coding studies? Some compromise in artistic integrity is to be expected when taking a live coder from a live venue and placing them in a sterile lab. It seems that ethnography remains the best means to capture the behaviour of a live coder in the wild. However, there might be some benefits to lab-based studies for live coders. Many live coders report composing ideas in the stillness of their own studios or houses, which they subsequently rework and reprogram live on stage. The creativity that occurs in front of an audience can also occur in a vacuum. But without the feedback loop that an audience provides, something will always be lost.

Perhaps the meaning of liveness embedded in the term ‘live coding’ is of a variable, ineffable quality. It can be somewhat pliant or viscous depending on the circumstances. As we have had to adjust to the post-pandemic environments, as the practice has shifted from small rooms with people close together to distributed and networked audiences in safer environments, it is conceivable that the definition of liveness can transform also.

2.1. Challenges and allure of liveness

The challenges faced by a live coder are similar but not entirely comparable with the challenges faced by other types of improvising musicians. Two main sources of challenges to the live coder can be identified: technological and cognitive. These categories can be subdivided to further understand live coders’ challenges. Some main technological issues can be classed as: hardware-based (e.g., laptop overheating during performance); software-based (e.g., bugs in the code); language-based (e.g., compile errors due to insufficient language tests); and network-based (e.g., latency in a network). Some main cognitive issues can be classed as: creative-based (e.g., unable to conceptualise a new pattern or sound); knowledge-based (e.g., the live coder feeling they are not ready to perform with their current knowledge); and algorithm-based (e.g., the live coder unable to understand how to execute a particular musical idea in algorithmic form).

While the risk posed by the possibility of either technological or cognitive issues is an aspect of most musical performances, the stakes are often greater with improvising in live coding practice. But the uncertainty and risk can hold attraction for both performer and audience. Indeed, crashing has always been an integral part of live coding’s (and the associated algorave scene’s) aesthetic (Armitage 2018). Armitage dissects the notion of how gender relates to the aspects of uncertainty and risk – in that women often experience feelings of imposter syndrome or simply the idea that it is beyond their

capacity, outside their interests and something that is intangible. Armitage also argues that creating spaces for women to learn and fail within and external to the core community is crucial for the development of a feminist movement for live coders.

2.2. Challenges and allure of live coding with machine agents

Collaboration with a machine agent can help broach some of the challenges faced in live coding, but can also pose new challenges for the performance that should also be considered. Figure 1 shows an example of such a collaborative session between a human agent and a machine agent. This session is in a prototyped text editor, modelled on similar collaborative editors such as Troop (Kirkbride 2017).

For the technical challenges discussed, when building the system designers could ensure the code they created is hypothetically executable by following the syntax of the language. For example, the TidalCycles agent built in Wilson, Lawson, McLean and Stewart (2021) can create syntactically correct patterns of code by leveraging Haskell’s strict type-system. However, this system does not yet have the same awareness of the code it produces that a human live coder might, and thus is more likely to produce code that can overload the SuperDirt synthesis engine. Awareness can be added using metrics such as weighting functions and arity; however, this only makes longer or more complex sequences less probable and does not remove them completely. Furthermore, if the agent is used in a networked system, this could further add network latency issues.

For the creative challenges discussed, one particularly useful case of using a machine agent is in combating the case of ‘conceptual un-inspiration’ (Wiggins and Forth 2018), that is, the parts of a performance where the live coder cannot produce novel conceptual ideas themselves. The contrast or complementation of the agents’ produced patterns of code could either provide some creative impetus for the live coder or drawback from the stylistic integrity of the performance for either the live coder or the audience experiencing it.

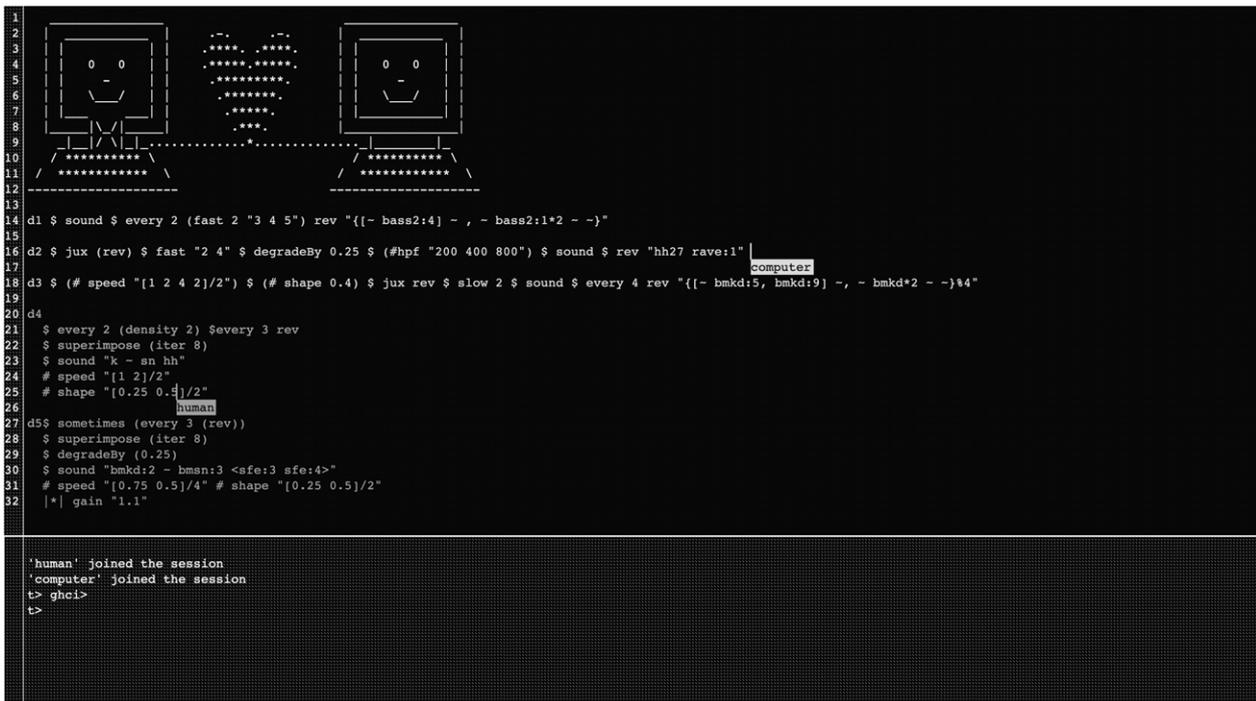
For the knowledge challenges discussed perhaps one key use case of an agent is for those who are trying to build their knowledge of the chosen languages. In allowing a better understanding of code sequences and syntactical relationships of the language, it might help build an overall understanding. Or further to this, rather than just language-specific examples, they may provide the human live coder with better knowledge of algorithmic thinking.

There may be a dimension of risk added to the performance using a machine agent. However, live

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 d1 $ sound $ every 2 (fast 2 "3 4 5") rev "[[- bass2:4] - , - bass2:1*2 ~ -]"
15
16 d2 $ jux (rev) $ fast "2 4" $ degradeBy 0.25 $ (#hpf "200 400 800") $ sound $ rev "hh27 rave:1"
17
18 d3 $ (# speed "[1 2 4 2]/2") $ (# shape 0.4) $ jux rev $ slow 2 $ sound $ every 4 rev "[[- bmkd:5, bmkd:9] -, - bmkd*2 ~ -)%4"
19
20 d4
21 $ every 2 (density 2) $every 3 rev
22 $ superimpose (iter 8)
23 $ sound "k - sn hh"
24 # speed "[1 2]/2"
25 # shape "[0.25 0.3]/2"
26
27 d5$ sometimes (every 3 (rev))
28 $ superimpose (iter 8)
29 $ degradeBy (0.25)
30 $ sound "bmkd:2 - bmsn:3 <sfe:3 sfe:4>"
31 # speed "[0.75 0.5]/4" # shape "[0.25 0.5]/2"
32 |*| gain "1.1"

```



```

'human' joined the session
'computer' joined the session
t> ghci>
t>

```

Figure 1. A prototype of a shared text editor session where a computer agent producing its own autonomous code patterns interacts with a human live coder. Collaborative editors such as Troop (Kirkbride 2017) exist for this challenge and this prototype is modelled on such interfaces.

coders are better equipped to deal with this due to this already being part of their practice. Further, Alperson (1984) notes that audiences are more forgiving of performances if they know that significant musical risks are taken by the performer. This added dimension of the risk in co-creative systems could install the same sense of sympathy and intrigue in audiences also.

Finally, the artists may be able to steer the inner workings of the agent – not only by building communication channels within the performance, but also by being able to modify the agent’s innards to ensure that its creative sensibilities perhaps match their own. Where corpus-based training is implemented as the generation source, then the live coder could either choose to work with an agent that matches their own live coding styles and preferences, or with something that works outside the boundaries of musical or coding practices they are used to.

3. THE ROLE OF REFLECTION

While reflection plays a key role in the creative process, it is an often ill-defined concept.

Generally, we might think of reflection to be defined as the notion of ‘sitting back’ and reviewing all or part of the written material, attempting to generate new ideas, making associations between cognitive

constructs, transforming existing ideas, and planning the future directions. The process of reflection can happen in a cyclical manner, with the creator oscillating between states of engagement and reflection (Sharples 1996). Both Baumer, Khovanskaya, Matthews, Reynolds, Schwanda Sosik and Gay (2014) and Ford and Bryan-Kinns (2022) propose a definition of reflection where at moments of uncertainty, different solutions are speculated upon. Key to their definition is the notion of uncertainty – and that reflection should be prompted at these points. Some of the research within the wider human–computer interaction (HCI) field also delves into reflection; for example, the discussion of reflection occurring both in-action and on-action (Schön 2017). Gaver, Beaver and Benford (2003) provide an alternative viewpoint, suggesting the design of intentionally suggestive artefacts to encourage reflection.

The question of what reflection could look like within a computationally creative system is addressed by first drawing a distinction between three types of creative systems: those that are purely generative, those that contain internal or external feedback, and those that are capable of reflection and self-reflection (Agres, Forth and Wiggins 2016). In the first case of purely generative systems, the creative system informs the creative process but not vice versa, and interaction between the generated artefacts and audience and the

creative system do not occur. In the second case of systems with internal or external feedback, the creative process can inform the creative system through this internal feedback and/or the audience and artefacts can provide external feedback to the system and its process.

The final and most sophisticated type in the hierarchy are those capable of reflection. Information regarding the creative process, generated artefacts, and/or the audience is taken into consideration by the creative system for self-assessment and reflection. It is this last element of reflection that is necessary for qualification as a truly creative system. The system must be able to reason about itself, either in reaction to feedback from the outside world or in light of internal evaluation processes. Some examples in the wider field of computer music that encourage reflection in systems can be found. Eigenfeldt (2007) produced work on designing multi-agent systems where the interacting agents, by listening and responding to each other, collaboratively develop compositional content. The behaviour of agents evolves over time, with agents 'reflecting' on their behaviour according to a number of pre-programmed personality traits.

One crucial aspect of the reflection process is the inclusion of models of human perception and cognition. Data from perceptual and physiological external evaluation methods can be used as the basis for the internal reflection process. For instance, the system may be able to anticipate how the artefacts it creates will affect its audience and whether it is possible to induce a particular effect from them (e.g., an affective state).

3.1. Reflection in live coding agents

The role of reflection in a live coding performance is crucial. One of the main roles of a live coder is exposing their cognitive processes through symbolic data. The live coder is constantly producing code, listening and then reflecting on the outcome. The transparency of reflection of a live coder may be present in the changes to code sequences they produce (e.g. they write a pattern and decide to make modifiable updates to match their musical intentions), or they may be obfuscated from the audience (e.g., they decide to make a shift in their conveyance of an intended affective state through different choices of samples, synthesis or rhythms).

A key part of any live coding performance is the fluctuation between the cognitive processes of engagement and reflection. Reflection or engagement states can be revealed to the audience due to whether code is being constructed, altered or deleted as well as physical clues from the performer. Perhaps one of

the most appealing aspects of the performance is watching the live coder struggle, reflect and then engage again; for example, when expressions on the performer's face fail to conceal that the music produced does not match its intentions. The same should apply with a machine agent that is incorporated into the performance. Its processes of reflection should be visible to the audience in the same way. In the lack of physical cues from the machine, the observable reflection could either come from watching it switch modalities from constructing patterns to modifying them, or even be made more opaque by having some chat function allowing the machine to output text to let the audience know which cognitive modality it was employing at any given time.

Reflection is particularly important for shifting towards the co-creative paradigm as it allows interaction that is continuously mediated through the performance. The live coder and machine agent gradually construct a shared, convergent meaning through each of their own processes of reflection. The machine agent's incorporation in the creative process also allows the live coder more time to reflect during the performance, especially if they usually perform alone and their subjective attention is often spread across multiple activities. Co-creative processes allow each to alternate between the engagement and the reflection states.

4. THE ROLE OF AESTHETICS

If we consider a goal of the machine to be the anticipation and induction of affect in the listener through reflection, we must consider where this model of affect comes from.

Music's ability to elicit strong and varied affective response from its listeners, performers and composers is a key part of its importance to us and within society. However, conducting research in the field can be demanding due to its inherent lack of quantifiable or measurable properties.

One of the biggest challenges presented in music and emotion research is building an understanding of the factors that contribute to imparting specific emotional connotation to music (Hevner 1935, 1937). Categorising and quantifying affective response is a challenging task and more so to interpret the effects of music within this. The expressive strategies adopted by performers and composers are not always obvious or accessible. The current literature addresses these challenges by a variety of means, all to different levels and measures of success. Juslin and Sloboda (2011) provide a wide, comprehensive viewpoint of the existing studies and research that deals with how musical factors map to expression.

4.1. Can a machine have an aesthetic preference?

Literature in the field of computational creativity has already begun actively considering the question of whether systems should be endowed with some sense of aesthetic preference to the generative material they are producing, rather than this being left to the human programming the system. Colton (2019) makes the case for this notion in the development of a seven-step roadmap for computational creativity. The second step deals with ‘appreciative systems’ wherein a system designer must encode aesthetic preference into a fitness function (*ibid.*), that is, endow a system with the ability to exclude from the possibility space based on some aesthetic rules. Colton then argues we should proceed along this trajectory for the third step: the level of ‘artistic systems’ where a system designer must ‘give the software the ability to invent its own aesthetic fitness functions and use them to filter and rank the images that it generates’ (*ibid.*).

This idea of expressive machines is surely alluring, but leads us towards parasocial relationships with our computers, expecting something from them that they are ill equipped to give us. We have already begun to see how well artificial intelligence can mimic art, but if we presuppose art as ‘the making of objects that are beautiful or express feelings’ then it has a relationship to the notion of expressivity similar to the protagonist of Searle’s Chinese Room. In summary, Searle’s argument imagines a person within a confined room who is unable to read or speak Chinese. That person has access to a wide amount of information, represented in terms of Chinese characters. This information is provided in a way that allows someone to use it by matching the characters together without deeper levels of understanding. In the same way, our machines are not capable of deeper levels of understanding musical meaning.

Additionally, feelings are often tied to their physiological sensations – one might describe the feeling of being excited as pulse-pounding, and for our computer this cannot make sense in the absence of a pulse. In arguing that the only true intelligence is in a body, Swafford (2022) describes this phenomenon succinctly:

When computers set out to do art, they don’t fashion it in a whirl of creative trance inflected by a deadline; they can’t account for the heat or alarming lack of it in the room, sensations in the groin, the failure or success of drawing a foot that looks planted on the ground, the failure or success of creating rhythmic momentum on the page, the bit that’s bullshit and needs to be fixed and the bit that’s really good and you see where it wants to go, the woman or man you just met who excites you and whom you hope to excite, the thought of the idiots who think they can write as well as you, also the bastards who write

better than you, what you’re having for dinner or what you had for dinner that’s not agreeing with you, the hairs falling out of your head onto the page, the expense of ink or paint or the rehearsal costs of a symphony orchestra, and so forth and so on.

We know that creative systems can be described in terms of their exploration of a conceptual space. If we refute the claim that machine-aesthetics are meaningless and accept them in any system we develop, then the method by which we generate them must be tied to a consciousness in the machine that we have yet to prove; or otherwise to accept that they must randomly be eliminating elements of the conceptual space in a detrimental way.

4.2. Aesthetics in live coding agents

Many systems proposing human–machine co-collaboration in live coding environments have still yet to consider the role of aesthetics in the collaborative process. In human–human collaboration this surely plays a vital role to the development of the piece through creating a sense of shared meaning, and it should be so in human–machine collaborations too.

The application of aesthetic theory into live coding practice is considered by Bell (2013) who proposes an aesthetic system based on Dewey’s ‘art-as-experience’ (Dewey 2008). Bell makes the distinction between affect, an affectee (i.e., a person experiencing affects in an interaction with affectors) and an affector (i.e., a percept that stimulates affects in an affectee). From this, the definition of an art experience arises as ‘the experience of affects in an affectee as the result of the affectee’s interaction with a network of affectors’ (Bell 2013: 4). Bell argues that live coding is experienced at many levels due to the assemblage of affectors. Different audiences will have varied reactions to the art experience, where audience members may come to the experience with disparities in code or musical knowledge. This variety of understanding of actions in the code may lead to novelty, excitement in one aspect of the code, whereas the same experience may result in frustration or confusion in another member.

As of yet, most agents integrated into live coding have focused more on the process of generation and not looked at the wider problem of how we might successfully collaborate with them in a human way. Bell’s work lays some foundations for this, and a few works have begun to discuss the importance of these issues (Wilson, Fazekas and Wiggins 2020; Xambó 2021) but a realisation of this in a collaborative and co-creative system has yet to be implemented. Section 6 considers speculative futures of the field and outlines a way that this could be realised.

5. THE ROLE OF EVALUATION

When considering the generation of live coded music, it is often useful to evaluate the music produced by the agent. This can be done either during the process or afterwards, and could be evaluated by 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 computationally creative systems do not always employ formal evaluation methods and many systems are not described in sufficient detail for their re-implementation. By identifying existing evaluation strategies and their relationship to the creative agency of a live coding machine agent, we propose an adapted model of evaluation for this specific case.

There exists research within the HCI field that deals specifically with either creative systems or improvisation, but evaluation of the intersection of these fields is more limited. Jordanous (2012) proposes a standardised procedure for evaluating creative systems (SPECs). Its approach is based around a set of 14 ‘components of creativity’ that evaluators should consider. Kantosalo and Sirpa (2019) also identify this disparity between the production and evaluation of creative systems and propose hybrid approaches from the field of user-experience design and computational creativity research.

The application of evaluation in the context of musical controllers (and their relationship to the task of improvisation) is considered by Kiefer, Collins and Fitzpatrick (2008). They focus particularly on strategies that are experience-focused rather than task-focused. The former are a part of a more recent movement in the area, while the latter are connected to traditional HCI. The authors employ both quantitative approaches using the statistical ANOVA measure and qualitative approaches using grounded theory of interview data in their evaluation metrics. Stowell, Robertson, Bryan-Kinns and Plumbley (2009) also use experiential approaches for the evaluation of live human-computer music-making. They contend that with live musical interactions, the traditional task-focused HCI methodologies such as talk-aloud protocols and task analysis, are not always suitable. Instead, they propose human-based comparative output analysis and discourse analysis as evaluation methodologies. Bernardo, Kiefer and Magnusson (2021) have also employed formal evaluation methodologies for Sema – a live coding environment aimed at supporting live coding with machine learning in the modern web browser. They employ the Creativity Support Index (CSI) to understand how well Sema supports creativity across its subsystems.

More recently, the topic of the critical issues in evaluating freely improvising interactive music systems has been discussed (Linson, Dobbyn and Laney 2012). The authors pose that it is crucial to ensure the suitability of the existing evaluation methods in order to make it possible for such systems to be studied scientifically. They make the case that, for some interactive computer systems such as those designed for freely improvised music, qualitative evaluation by experts is the most appropriate evaluation method. Hodson (2017) suggests the research field of computational creativity has committed a ‘fundamental misunderstanding’ by assuming that creativity is an *ex ante* phenomenon (i.e., assuming that creativity is caused by certain cognitive processes) rather than an *ex post* phenomenon (i.e., being an observable property in context when it happens, no matter how it is produced). Wiggins (2021) also proposes that the *ex ante* reading of creativity is erroneous and that many researchers search for systems with characteristics that enable novel and valuable outputs, which may only be judged creative in the *ex post* condition.

5.1. The CAT as a tool in the evaluational arsenal

There are many aspects to explore when considering the evaluation of human-machine collaborative systems in live coding. Live coding encompasses many parts that can be evaluated: the sonic quality of the music created, the quality of the code produced, the audience perception of both the music and the experience and so on. When adding agents into the process, more dimensions to evaluation processes are introduced; for example, not only whether it adds to the musical experience but also how the performer feels while collaborating with the agent – do they enjoy it, do they struggle to understand what is going on and why it is doing certain things. McCormack, Gifford, Hutchings, Llano Rodriguez, Yee-King and d’Inverno (2019) complete such an evaluation in their collaborative human-AI improvisational system, which looks at establishing trust in these partnerships by looking to communicate internal states.

Because of the vastness of these challenges, this article cannot hope to cover all these facets of evaluation. The HCI-aspects of evaluation are not considered here but are certainly pertinent to future discussions of these sorts of collaborations. Instead, we look particularly at how to evaluate whether the inclusion of such a partnership makes a difference to the perception of creativity, and whether there are any significant changes.

We propose a model for one particular facet of evaluation of creative agents in live coding taking Hodson’s (2017) argument of treating creativity as an *ex post* phenomenon into account. This conceptually

allows for creativity and inspiration to be found anywhere in the system. For this, the consensual assessment technique (CAT) developed by Amabile (1982) is modified for usage for those looking to develop computationally creative systems in live coding.

Amabile laid the case for the consensual assessment technique by noting that experimental studies for social and environmental influences on creativity were rare, stating a major obstacle to the research of the criterion problem – the lack of a clear operational definition and appropriate assessment methodology. The CAT offers a methodological approach to evaluating computational models of musical compositions that supports the view of creativity as an *ex post* phenomenon. Amabile presents a consensual definition of creativity and a reliable subjective assessment technique based on the definition is described:

A product or response is creative to the extent that appropriate observers independently agree it is creative. Appropriate observers are those familiar with the domain in which the product was created or the response articulated. Thus, creativity can be regarded as the quality of products or responses judged to be creative by appropriate observers, and it can also be regarded as the process by which something so judged is produced. (Amabile 1982: 1001)

Amabile's Consensual Assessment technique provides a framework with which we can frame evaluations. However, there are some notable drawbacks with Amabile's proposed evaluation methodology. Some differences include that Amabile's work treated creativity in the broadest sense and Amabile's domain did not deal with improvisation in music in the same way. Although the approach was used for more general views on how creativity can appear, the framework can still be useful for the live coder, as the creativity is not treated as happening a priori, but instead in the manner that Hodson (2017) suggested.

To attempt to adapt the CAT as a useful evaluation tool for building systems with some modelling of creative behaviour, we consider the work of Boden (2004) on what creativity means. Boden's view of creativity is underpinned by the connection between creativity and the novelty and value of the artefacts. Incorporating this into the evaluation process would involve a modification of the experimental tool to be used in a way that formally applies this definition.

The adapted CAT could be structured as follows. The live coder is asked to create a performance with the machine agent (and without the agent for a control). Section 2 justifies this modification of the typical 'live' performance to a more controlled environment, by examining how the definition of live for live coders has been altered due to the COVID-19

pandemic. This can then be captured by means of audio and/or video performance and sent blind to the carefully selected panel of experts in the field. The experts might not necessarily be live coders themselves, but should have a deep knowledge of the related fields and intersections that it explores, and most importantly, be well trained in being able to articulate their thoughts around such matters. The ability to articulate is crucial as this is where the evaluation lies, in their ability to explain if they feel creativity is there and exactly what about it is making them think so.

By employing this adapted CAT for live coders, any performance between the computer agent and human live coder would thus be improved by moving towards Boden's model of creativity. The evaluation of its artefacts, by a group of experts, would allow us to examine these properties of novelty and value within the artefacts themselves. Or perhaps examining other dimensions that might confound, such as where technical skill could be conflated with creative ability. Finally, another benefit of this particular method of evaluation would be that the anonymity could limit potential biases that might otherwise exist in this type of evaluation; for example, assumptions made about performers, or for those who are less inclined to believe in the creative potential/autonomy of machines.

6. FUTURE CONSIDERATIONS

Live coding has been at the forefront of human-machine interaction for the past few decades, and has advocated for the creative rights of both human and computer within this partnership. The act of writing generative algorithms live can shape the human's creativity and push them in directions they might not conventionally explore. The allure of risk, and the creative opportunities prompted by error (and the political philosophies that underlie this) have been integral parts of the creative process. The integration of machine agents into the creative process is an extrapolation of the foundations laid by the community and with the growing interest in fields of artificial intelligence and computational creativity, it is clear that a precedent is set for machine agents in live coding.

To close, speculative futures of the integration of live coding agents are discussed, which adhere to the principles of reflection, aesthetics and evaluation outlined herein. These considerations are language-agnostic, but assume some properties of a live coding language, such as the artist-programmer's ability to construct basic music extensionally (i.e. in literal notes or samples) and then elaborate on it by a mixture of additional extensional structures (i.e., adding a

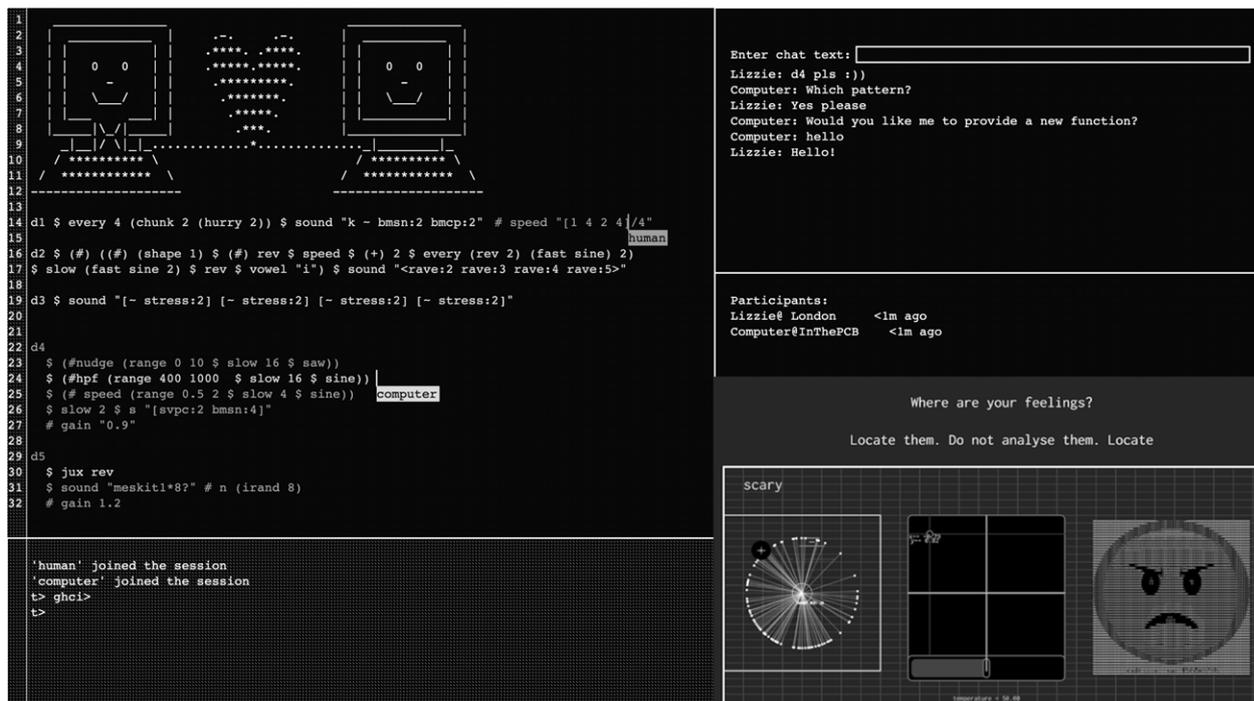


Figure 2. A speculative design of an interface for human–machine co-creative live coding, building on the design of Figure 1 to incorporate elements of aesthetics, reflection and evaluation.

polyrhythmic patterns) or intentionally (i.e., adding other functions to manipulate the pattern). We do not consider the case of the machine agent ‘replacing’ the human, but instead outline how co-creation is best facilitated for live coders.

A live coding agent imbued with the property of reflection could perform a variety of functions within the performance. The first is in prompting the live coder at moments of their reflection. For the definition of reflection discussed in section 3 – as solutions being speculated upon at moments of uncertainty – then the agent could be useful for exploring certain aspects of a conceptual space that the human live coder have not been able to conceptualise themselves. The patterns produced by a machine agent imbued with the capability of reflection would help mediate the performance of both live coder and machine.

A live coding agent with the ability to make aesthetic decisions based on human models of perception and cognition could be used to shape the progression of the piece. The human agent could perhaps choose an aesthetic value for the machine that either aligned closely to their own or perhaps they might choose something particularly antithetical, to work against a disruptive agent in performance. In some senses, live coding is particularly suited to the task of collaboration with a machine agent. Creating music through code displays the dialogue between the machine and human agents in the system in a way that

can be meaningful for both the parties themselves and the audience observing.

For co-creative systems, humans are often required to take on the role of evaluator, but this paradigm can also be flipped, where the machines are given the task to evaluate the human. Indeed, some research in this field has begun to take place; for example, Collins and Knotts (2019) leverage the capabilities of machine listening in JavaScript to complete such a task. We have established the role that aesthetic considerations play in the co-creative process, and where these should be sourced from with the machine agent. Perhaps if we imbued the machine with the ability to evaluate the music created by humans we could gain new insight and would move towards a more collaborative partnership. Moreover, human evaluators are unreliable and subject to many kinds of bias. Machines are also likely to pick up the biases of humans embedded in datasets, but perhaps the design of the evaluator could also be influenced by fields of affective computing, such as the ‘empathetic machine’ (Fung et al. 2016; Fung, Bertero, Xu, Park, Wu and Madotto 2018). These empathetic evaluators could push the live coder to augment their practice while maintaining a positive relationship.

Figure 2 shows a speculative design for a potential interface for a co-creative live coding agent to collaborate with a human live coder, which extends the simple collaborative interface in Figure 1. In this,

we can see at the bottom there is a chat function, based on that found in the Estuary collaborative editor (Ogborn, Beverley, del Angel, Tsabary, McLean and Betancur 2017). This chat function allows the live coder to communicate with the machine. It could then be used to accept the computers' evaluation of their pattern and provide alterations to their code, as well as creating its own, in a manner similar to how collaborative human live coders might interact. A model of affective response is also incorporated in, using the valence-arousal model (Russell 1980) where valence (x -axis) represents the concept of how positive or negative the affective state is and the arousal (y -axis) represents how high or low energy it is. This would allow control from the live coder, and the ability to model aesthetics on the basis of the human model. Overall, this speculative interface between human and machine agent provides the framework for some of the ideas discussed herein.

Acknowledgements

This work was supported by EPSRC and AHRC under the EP/L01632X/1 (Centre for Doctoral Training in Media and Arts Technology) grant. G. Wiggins received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen".

REFERENCES

- Agres, K., Forth, J. and Wiggins, G. A. 2016. Evaluation of Musical Creativity and Musical Metacreation Systems. *Computers in Entertainment (CIE)* **14**(3): 1–33.
- Alpers, P. 1984. On Musical Improvisation. *The Journal of Aesthetics and Art Criticism* **43**(1): 17–29.
- Amabile, T. M. 1982. Social Psychology of Creativity: A Consensual Assessment Technique. *Journal of Personality and Social Psychology* **43**(5): 997–1013.
- Armitage, J. 2018. Spaces to Fail in: Negotiating Gender, Community and Technology in Algorave. *Dancecult: Journal of Electronic Dance Music Culture* **10**(1). <http://doi.org/10.12801/1947-5403.2018.10.01.02>.
- Baumer, E. P., Khovanskaya, V., Matthews, M., Reynolds, L., Schwanda Sosik, V. and Gay, G. 2014. Reviewing Reflection: On the Use of Reflection in Interactive System Design. *Proceedings of the 2014 Conference on Designing Interactive Systems*. New York: ACM, 93–102.
- Bell, R. 2013. Towards Useful Aesthetic Evaluations of Live Coding. *Proceedings of the 2013 International Computer Music Conference*. Perth: ICMA, 236–241.
- Bernardo, F., Kiefer, C. and Magnusson, T. 2021. November. Assessing the Support for Creativity of a Playground for Live Coding Machine Learning. *International Conference on Entertainment Computing*. Cham: Springer, 449–56.
- Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*. 2nd Edition, Routledge, London.
- Collins, N. and Knotts, S. 2019. A JavaScript Musical Machine Listening Library. *Proceedings of the 2019 International Computer Music Conference*. San Francisco: ICMA, 383–87.
- Colton, S. 2019. From Computational Creativity to Creative AI and Back Again. *Interalia Magazine*.
- Dewey, J. 2008. *Art as Experience*. London: Penguin.
- Eigenfeldt, A. 2007. The Creation of Evolutionary Rhythms within a Multi-Agent Networked Drum Ensemble. *Proceedings of the 2007 International Computer Music Conference*. Copenhagen: ICMA, 267–70.
- Ford, C. and Bryan-Kinns, N. 2022. Speculating on Reflection and People's Music Co-Creation with AI. *Generative AI and HCI Workshop at CHI 2022*, 10 May. <https://qmro.qmul.ac.uk/xmlui/handle/123456789/80144>.
- Fung, P., Bertero, D., Wan, Y., Dey, A., Chan, R.H.Y., Bin Siddique, F., et al. 2016. Towards Empathetic Human-Robot Interactions. *International Conference on Intelligent Text Processing and Computational Linguistics*. Cham: Springer, 173–93.
- Fung, P., Bertero, D., Xu, P., Park, J. H., Wu, C. S. and Madotto, A. 2018. Empathetic Dialog Systems. *The International Conference on Language Resources and Evaluation*. European Language Resources Association.
- Gaver, W. W., Beaver, J. and Benford, S. 2003. April. Ambiguity as a Resource for Design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York: ACM, 233–40.
- Hevner, K. 1935. The Affective Character of the Major and Minor Modes in Music. *The American Journal of Psychology* **47**(1): 103–18.
- Hevner, K. 1937. The Affective Value of Pitch and Tempo in Music. *The American Journal of Psychology* **49**(4): 621–30.
- Hodson, J. 2017. The Creative Machine. *Proceedings of the International Conference on Computational Creativity 2017*. Atlanta, GA: ACC, 143–50.
- Jordanous, A. 2012. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What It is to be Creative. *Cognitive Computation* **4**(3): 246–79.
- Juslin, P. N. and Sloboda, J. A. 2011. *Handbook of music and emotion: theory, research, applications*. 2nd Edition, Oxford University Press, Oxford.
- Kantosalo, A. and Sirpa, R. 2019. Experience Evaluations for Human-Computer Co-Creative Processes – Planning and Conducting an Evaluation in Practice. *Connection Science* **31**(1): 60–81.
- Kiefer, C., Collins, N. and Fitzpatrick, G. 2008. HCI Methodology For Evaluating Musical Controllers: A Case Study. *Proceedings of the 8th International Conference on New Interfaces for Musical Expression*. Genova: ACM, 87–90.
- Kirkbride, R. 2017. Troop: A Collaborative Tool for Live Coding. *Proceedings of the 14th Sound and Music Computing Conference*, Finland: ICMA, 104–9.
- Linson, A., Dobbyn, C. and Laney, R. C. 2012. Critical Issues in Evaluating Freely Improvising Interactive Music Systems. *Proceedings of the International Conference on Computational Creativity 2012*. Dublin: ACC, 145–9.
- McCormack, J., Gifford, T., Hutchings, P., Llano Rodriguez, M. T., Yee-King, M. and d'Inverno, M.

2019. In a Silent Way: Communication between Ai and Improvising Musicians beyond Sound. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, 1–11.
- Merriam-Webster. 2022. Online Dictionary. www.merriam-webster.com/dictionary/live (accessed 11 August 2022).
- Ogborn, D., Beverley, J., del Angel, L. N., Tsbary, E., McLean, A. and Betancur, E. 2017. Estuary: Browser-based Collaborative Projectional Live Coding of Musical Patterns. *International Conference on Live Coding (ICLC)*, Morelia, Mexico.
- Russell, J. A. 1980. A Circumplex Model of Affect. *Journal of Personality and Social Psychology* **39**(6): 1161.
- Schön, D. A. 2017. *The Reflective Practitioner: How Professionals Think in Action*. Abingdon: Routledge.
- Sharples, M. 1996. An Account of Writing as Creative Design. In C.M. Levy and S. Randell (Eds.) *The Science of Writing*. Mahwah, NJ: Erlbaum, 127–48.
- Stowell, D., Robertson, A., Bryan-Kinns, N. and Plumbley, M. D. 2009. Evaluation of Live Human–Computer Music-Making: Quantitative and Qualitative Approaches. *International Journal of Human-Computer Studies* **67**(11): 960–75.
- Swafford, J. 2022. The Intelligence of Bodies. *VAN Magazine*. <https://van-magazine.com/mag/jan-swafford-beethoven-x/> (accessed 16 September 2022).
- Tanimoto, S. L. 2013. A Perspective on the Evolution of Live Programming. *2013 1st International Workshop on Live Programming (LIVE)*. San Fransisco, CA: IEEE, 31–4.
- Wilson, E., Fazekas, G. and Wiggins, G. 2020. Collaborative Human and Machine Creative Interaction Driven through Affective Response in Live Coding Systems. *International Conference on Live Interfaces*, Trondheim, Norway.
- Wilson, E., Lawson, S., McLean, A. and Stewart, J. 2021. Autonomous Creation of Musical Pattern from Types and Models in Live Coding. *Proceedings of the 9th Conference on Computation, Communication, Aesthetics & X*. Porto, Portugal, 76–93.
- Wiggins, G. A. 2021. Creativity and Consciousness: Framing, Fiction and Fraud. *Proceedings of the International Conference on Computational Creativity 2021. Mexico: ACC*, 182–91.
- Wiggins, G. A. and Forth, J. 2018. Computational Creativity and Live Algorithms In Dean, R. T (Ed.) *The Oxford handbook of algorithmic music*. Oxford University Press, Oxford.
- Xambó, A. 2021. Virtual Agents in Live Coding: A Short Review. *arXiv preprint arXiv:2106.14835*. <https://doi.org/10.48550/arXiv.2106.14835>.