The Liveness of Live Coding: A Musical Perspective on the Many Livenesses

Georgios Diapoulis
University of Gothenburg and Chalmers University of Technology
georgios.diapoulis@chalmers.se

ABSTRACT

What is the liveness of live coding and why should we care about it? In this article I discuss how we came to use the term in live coding and how the term is framed across different research areas and artistic practices. This discussion forefronts some tensions but also shared understandings. I provide an interpretation of liveness in the context of musical live coding and discuss how it converses with the many interpretations of the liveness(-es). A shared understanding emerges among electronic musicians about the different qualities of liveness that extends beyond the technical aspects of it.

1 Introduction

Ten years after the first ICLC, there is increasing interest in the liveness of live coding. A technical explanation of liveness in programming environments was provided by Tanimoto (2013) during the Live Programming Workshop, and since then, the many liveness(-es) have been presented in the Live Coding User's Manual (Blackwell et al. 2022). The *liveness of live coding* is conceived as something in between the human and the machine, where the boundaries between autonomous processes and human actions are blurred. Electronic music is an exemplar case of machine autonomy, and this autonomy becomes evident when performing with electronic musical interfaces. Humans follow mental processes to compose the textualities of code, and machines execute the written code to perform actions perceived by humans — or we may simply attribute these actions to the machines. Regardless of any ontological questions related to the performativity of machines, a pragmatic approach would be to include these "virtual actions" in a larger category of performed actions (if we avoid any discussion about the intentionality of machines). Even if these virtual actions belong to our mental imagery, they are still capable of causing neural activations, which in turn induce subjective percepts.

Action-oriented ontologies for electronic music performance highlight an agential understanding. It is this characteristic machine autonomy that can induce a sense of 'otherness' in the machine. While the distinction between an actor and a spectator may not be of paramount importance in the context of live coding, it is certainly a question worth posing. For some, live coding without an audience is not truly live coding (Blackwell and Collins 2005), whereas for others, art without an audience opens possibilities for public thinking (McLean, Rohrhuber, and Wieser 2023).

In this article, I will discuss common patterns of liveness that emerge across different research disciplines, such as software engineering, the psychology of programming, electronic music, media studies, and performance studies. I will present my understanding of liveness in the context of musical live coding and identify an overarching theme that would lead me to present a conceptual framework of liveness. I will then raise questions about the futures of the many liveness-es within a live performance and discuss this in the context of interactive AI technologies.

It is often the case that we may overlook the technical dimension of liveness as was introduced by Tanimoto (1990, 2013), or simply the system's perspective. On the other hand, liveness is a fluid concept for Auslander (2022), and from a cognitive perspective it should include experiential and perceptual connotations. For instance, are there different experiences between the performer's and audience's perceptions of liveness? How does liveness converse with the perception of continuity, also known as chunking (Snyder 2000), when live coding? How lively is live coding when using a keyboard in comparison to playing a musical instrument? These are some of the questions this article will tackle.

2 Liveness: Aesthetics, Music Interaction, and Software Engineering

Liveness has been discussed in a variety of disciplines, such as software engineering (Tanimoto 1990, 2013), human-computer interaction (Church, Nash, and Blackwell 2010; Nash and Blackwell 2012), musical aesthetics (Croft 2007; Emmerson 2017; Di Scipio 2021), media theory (Auslander 2008) and performance studies (Norman 2016). In live coding music performance, liveness is an indispensable part. Here, I attempt to shed light on what liveness is, how it is applied in musical live coding systems and how liveness exhibits different facets.

2.1 Liveness in Electronic Music

Emmerson (2017) and Croft (2007) discuss liveness in the context of electronic music. Emmerson addresses several aspects of embodiment based on the relationships between the performer's gestures and audience perception. On the other hand, Croft views electronic music as an inherently disembodied activity and presents his perspective on embodiment in electronic music by establishing a foundation in traditional instrumental musicianship. Some of Croft's "conditions for instrumentality" (ibid.) include the proportionality of the performer's actions to the computer's responses, energy relations between the performer's actions and the system's responses, synchronicity of onsets, and stable mappings, to name a few. I call this *perceived liveness*, as it aims to establish a common ground for both performers and audiences. However, the conditions posed by Croft do not provide further insights, in the context of electronic music, into reducing what Leman (2007) calls indirect involvement. Indirect involvement presupposes a symbolic mediator between the subject (musician) and the musical outcome, where, in the case of live coding, this symbolic mediator is the written code. Computer code, as notational segments, is the most widespread form of live coding. It is also true that "there is no natural mapping between gesture and sound in digital systems" (Magnusson 2009), with only a few instances of embodied metaphors, such as pitch height and gestures. Leman observes that these symbolic descriptions mediate music in a way that leads to indirect involvement, which can hinder the embodied experience of both the performer and the listener. He suggests that digital musical instruments and electroacoustic devices can reduce this perceptual distance. Another proposal to overcome such difficulties —- in the context of live coding —- can be the bodily involvement of the performer's gestures, as seen in the case of CodeKlavier (Noriega and Veinberg 2019), where the pianist is live coding while playing the piano.

2.2 Liveness in Psychology of Programming and Software Engineering

Nash and Blackwell (2012) have analyzed the "perceptions of liveness" in computer music software, as these perceptions are mediated by notation. They provide quantitative measures of the flow components with the cognitive dimensions of notation (CDs). Liveness, in this sense, is a domain feedback entity (Tanimoto 2013). Flow components refer to the concept of "flow" as introduced by Csikszentmihalyi, where phenomenological approaches to optimal experiences are examined (Nakamura, Csikszentmihalyi, et al. 2002). The CDs are information structures that are characteristic of programming environments. Visibility, progressive evaluation, and consistency have proven to be the most prominent CDs. From the flow dimensions, intrinsic reward is the most prominent and is correlated with progressive evaluation, which is a measure of domain feedback. The progressive evaluation of code chunks is at the heart of live coding practices. The study by Nash and Blackwell (2012) is focused on edit-audition feedback cycles in digital audio workstations (DAW), and concludes that trackers, and generally the tape recorder metaphor, reduce the perception of liveness when using visual notations.

Tanimoto (1990) introduced the term *liveness* in software engineering with his hierarchy of liveness for programming environments. As a jazz pianist, Tanimoto was inspired by live music and introduced liveness as a technical term to communicate advancements in visualization technologies for visual programming. I call this *technical liveness*. He presented a hierarchy of liveness with four distinct levels: informative (L1), informative and significant (L2), informative, significant, and responsive (L3), and informative, significant, responsive, and live (L4). Later, as part of a symposium on live programming, Tanimoto revisited his hierarchy of liveness in 2013 and introduced two additional levels: level-5 liveness, also known as tactically predictive (L5), and level-6 liveness, also known as strategically predictive (L6). A tactically predictive system can inform users about their programming behaviors, whereas a strategically predictive system adds steps in reasoning. In this article, I will refer to L5 and L6 as advanced levels of liveness.

2.3 Liveness in Musical Live Coding

As discussed in the previous section, liveness in programming environments depends on the notation of the language and the environment itself (Church, Nash, and Blackwell 2010). Notation typically consists of the functional parts of code chunks in live coding. However, secondary notation, such as comments, indentation, and syntax highlighting, can

also play an essential role in dramatizing an algorithm (Collins 2011). As an environment, we consider both the code and the auditory representation for the listeners. Although some authors view the audience as end-users who consume code (Blackwell and Collins 2005), here, "listeners" refer to coders and audiences. The coder establishes constancy in the programming environment's interactions using auditory percepts. Audition is of utmost importance in musical live coding, as it can be impossible to imagine the generated sounds without listening to the audio output, even when sound visualizations, such as spectrograms, are involved (Diapoulis 2022). As long as the running program is rendered to sound within a performance context, the live coding system is necessarily classified as L4. The code is rendered to another modality (sound/musical output), and an auditory stream (listeners' perception) facilitates perceptual continuity between code segments and the musical outcome. Thus, the coder's musical perception establishes a feedback loop between the code and the sound. I call this *felt liveness*, as another modality (audition) mediates to establish this relationship between code and sound. Thus, the term indicates how we experience liveness.

An L5 system should be able to predict and run near-future commands while providing the user with alternatives. Recently, two independent cases of co-creative systems were presented, offering a preliminary understanding of what an L5 system may look like during a live coding performance (Attanayake et al. 2020; E. Wilson et al. 2021). Essentially, both systems make predictions based on patterns of code written by the performer and suggest future alternatives. In Attanayake's design, the authors experimented with a disruptive mode in which code modifications were executed without the user's permission. While a 'thinking ahead' feature may be indirectly acknowledged as an unwelcome aspect during a live coding session (Roberts and Wakefield 2018), Attanayake's disruptor mode was perceived as more enjoyable during the user study, demonstrating a playful and creative approach to using such predictive models.

Elizabeth Wilson, Fazekas, and Wiggins (2023) presented two main challenges of liveness during a live coding session: technical challenges that need to be addressed and cognitive challenges that must be compensated for. Elizabeth (Lizzie) Wilson is a highly active performer, and I expect this view to resonate with many live coders. It is certainly the case that, on the system side, several things can be unclear to the coder, and during a live performance, a wide range of factors may be at stake, potentially leading things in the wrong direction. Additionally, a large amount of cognitive resources is required for both writing and editing code, along with a keen ear to ensure the pace and flow of the session.

3 Cognition, Living Systems and Metaphors We Code By

In this section, I discuss liveness from a cognitive point of view. I will continue with how control is expressed in live coding and why it may be important or not. I will close this section with my perspective on "flow and pace," which could be seen as a brief follow-up to Roberts and Wakefield (2018) view on the topic.

3.1 Cognition

Here cognition is seen as the making of bodily and brain activities. The brain orchestrates bodily movements, and in response, our bodily percepts influence our neural activity. Under these lenses, cognition involves both experience and perception. Experience is typically subjective, whereas perception can be reproducible within certain populations. For instance, if I am synesthetic, I may have a unique color experience when smelling cinnamon buns. On the other hand, most people can identify the odor of cinnamon and recognize it when they participate in a forced-choice perceptual study.

How is cognition involved during a live coding session? I support the view that human cognition encompasses both subjective experiences and percepts that may be shared between performers and audiences. As liveness is an indispensable part of a live coding session, human cognition is in constant interplay with machine actions. Electronic music is well known for conserving a degree of autonomy, thus highlighting the blend between human cognition and machine intelligence. As mentioned before, who is doing what may not be of particular importance; rather, it is the combined outcome of both human and machine that matters.

Di Scipio (2021) argues that cognition can emerge from minimal agents and addresses liveness as a biological metaphor. When we move beyond a human perspective, we assign the quality of 'live' to an algorithm, program, circuit, or other entity. The view on minimally cognitive systems is a well-known example in biology, as we may observe minimal agents that, as a whole, exhibit emergent properties that cannot be traced back to the individual units. It is an important point of view, as it shifts the perspective from instruments to environments.

3.2 Control

Control in live coding is certainly a concern for some live coders, like Sam Aaron (2018). It is not uncommon to see a performance where the live coder demonstrates control over the session, with perfectly orchestrated code chunks built

in a progressive manner. At the same time, the generativity of the possible spaces that emerge through a 'blank slate' session may be impossible to tame. Control in such a scenario may depend solely on the experience and fluency of the performer, although the programming environment can also assist (e.g., using Sonic Pi, which was specifically developed to foster programming education). When this occurs in a networked environment, temporal uncertainty comes into play (Rohrhuber and Campo 2004), as no one can predict network delays, different realizations of a performance, and the multiplicities of musical outcomes. In such multiple worlds scenarios, understanding what constitutes a musical composition becomes challenging to perceive, document, archive, and reproduce.

When a "system of action" is in control, a "letting go of control" becomes necessary (Blackwell et al., 2022, p. 167). No one can predict what will happen; no one can even perceive what is actually occurring. Everyone is experiencing a slightly (or largely) different world. We can imagine an intercontinental laptop ensemble where different audiences have radically different experiences regarding the aesthetics of the musical outcome. Why is this interesting in the first place? A naive technological answer might be: why is it not interesting to experience the displacements of the IT infrastructure (e.g., network delays), as long as such technologies enable us to convey our voices, facial expressions, and shared living experiences over thousands of miles? Don't we already experience slight delays with web conferencing tools? Why not expose such system flaws to create art? Of course, such networked art practices and philosophical investigations are not new; they have several decades of history, from Pauline Oliveros with her collective performances over telephone lines to The Hub. Inspired by second-order cyberneticians, we can conceptualize these uncertainties during a live coding session as a controlled scenario within an uncontrolled environment.

An interesting point of juxtaposition is that of liveness and mediatization, as noted by Auslander (2008), though he later considers this tension to be less significant (Auslander 2022). Imagine a scenario where the same TV program is being broadcast from America to the rest of the world. We would expect that a viewer in America would experience a slight delay until their antenna receives the signal, a viewer in Europe would experience a slightly larger delay, and a viewer in Asia would experience the largest delay. What is common among the three viewers is that all of them will be watching the exact same TV pictures, listening to the exact same words, and so on. Now, imagine a networked laptop ensemble with three performers across America, Europe, and Asia. Depending on their systems' setups, they might be experiencing radically different versions of the performance. When liveness is viewed in the context of live coding, I believe the tension between mediatization and liveness becomes evident.

3.3 Flow and Pace

There is an anticipated tension between advanced levels of liveness (i.e., L5, L6) and flow and pace in live coding. In Roberts and Wakefield (2018), interviewees reported that planning ahead is a cognitively demanding task for the user. In Attanayake et al. (2020), a trade-off between sophistication and efficiency is acknowledged. Such a trade-off may suggest that virtual agents for code previews could be avoided by users, despite the agents' creativity appraisal. On the other hand, Lizzie Wilson sees that predictive technologies that blend human and machine agencies can provide users with the space to explore cognitively demanding tasks (Blackwell et al., 2022, p. 171). It is certainly the case that in any electronic music practice, where autonomous processes are ubiquitous, the user can set their own pace when making decisions, unlike in traditional instrumental performance.

4 Common Themes and Tensions Among the Many Livenesses

In media studies, liveness is often perceived as a live transmission, akin to a TV or radio broadcast (Auslander 2022; Couldry 2004). In live coding, however, we tend to start from a technical foundation of liveness, as presented by Tanimoto (2013), prioritizing interactivity with the programming environment. There is a substantial bibliography on liveness and mediatization, with Auslander having published three editions of his book on "Liveness." In his earlier work, Auslander (2008) identified a tension between the concepts of mediatization and liveness. Blackwell et al. (2022) (p. 162) discuss liveness in an ontological context. In contrast, the later writings of Auslander (2022) suggest that centering the tension between liveness and mediatization in an ontological framework may be unproductive. This tension may be explained by the fact that both "Live Coding: A User's Manual" and the third edition of "Liveness" by Auslander were published in the same year, 2022.

Albeit the many different views on liveness, I identify common themes among electronic musicians and live coders. For instance, Di Scipio (2021) outlines three main components for an agential understanding of liveness: human, technical, and environmental. I have presented the liveness of live coding as a three-fold concept comprising perceived, felt, and technical liveness (Diapoulis 2023). The term perceived liveness includes an environmental component by means of audience perception. Furthermore, my perspective on liveness aligns with the view expressed by Elizabeth Wilson, Fazekas, and Wiggins (2023), highlighting that liveness encompasses both cognitive and technical dependencies. In that

sense, cognition, as discussed earlier, involves a combination of perception and experience. At a high level of abstraction, there is a shared understanding among electronic musicians regarding what liveness is and how we experience it.

As for the future, it would be interesting to see whether and how the concept of liveness can be transferred to other domains, such as real-time machine learning and machine listening. Is it plausible to foresee what a predictive model will learn in real-time? Probably not, as one cannot learn what is not known. However, in the context of machine listening, I can imagine previewing features (Collins 2015). If that is the case, then liveness becomes a cross-domain feedback entity on the technical level, something I am advocating in the present article between the cognitive and the technical domains.

References

Aaron, Sam. 2018. "Live Coding Education." *The MagPi Educators Edition*, 44–47.

Attanayake, Ushini, Ben Swift, Henry Gardner, and Andrew Sorensen. 2020. "Disruption and Creativity in Live Coding." In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 1–5. IEEE.

Auslander, Philip. 2008. "Live and Technologically Mediated Performance." *The Cambridge Companion to Performance Studies*, 107–19.

———. 2022. *Liveness: Performance in a Mediatized Culture*. Routledge.

Blackwell, Alan F, Emma Cocker, Geoff Cox, Alex McLean, and Thor Magnusson. 2022. *Live Coding: A User's Manual*. MIT Press.

Blackwell, Alan F, and Nick Collins. 2005. "The Programming Language as a Musical Instrument." In *PPIG*, 11.

Church, Luke, Chris Nash, and Alan F Blackwell. 2010. "Liveness in Notation Use: From Music to Programming." In *PPIG*, 2.

Collins, Nick. 2011. "Live Coding of Consequence." *Leonardo* 44 (3): 207–11.

———. 2015. "Live Coding and Machine Listening." In *Proceedings of the International Conference on Live Coding*, 4–11.

Couldry, Nick. 2004. "Liveness,'reality,' and the Mediated Habitus from Television to the Mobile Phone." *The Communication Review* 7 (4): 353–61.

Croft, John. 2007. "Theses on Liveness." *Organised Sound* 12 (1): 59–66.

Di Scipio, Agostino. 2021. "Thinking Liveness in Performance with Live Electronics." *Sound Work: Composition as Critical Technical Practice*, 171.

Diapoulis, Georgios. 2022. "Livecode Me: Live Coding Practice and Multimodal Experience." In *Proceedings of Psychology of Programming Interest Group*. PPIG2022.

———. 2023. *Expression in Live Coding: Gestural Interaction for Machine Musicianship*. Chalmers Tekniska Hogskola (Sweden).

Emmerson, Simon. 2017. *Living Electronic Music*. Routledge.

Leman, Marc. 2007. *Embodied Music Cognition and Mediation Technology*. MIT press.

Magnusson, Thor. 2009. "Of Epistemic Tools: Musical Instruments as Cognitive Extensions." *Organised Sound* 14 (2): 168–76.

McLean, Alex, Julian Rohrhuber, and Renate Wieser. 2023. "The Meaning of Live: From Art Without Audience to Programs Without Users." In *The Seventh International Conference on Live Coding*.

Nakamura, Jeanne, Mihaly Csikszentmihalyi, et al. 2002. "The Concept of Flow." *Handbook of Positive Psychology* 89: 105.

Nash, Chris, and Alan F Blackwell. 2012. "Liveness and Flow in Notation Use." In *NIME*.

Noriega, Felipe Ignacio, and Anne Veinberg. 2019. "The Sound of Lambda." In *Proceedings of the 7th Acm Sigplan International Workshop on Functional Art, Music, Modeling, and Design*, 56–60.

Norman, Sally Jane. 2016. "Senses of Liveness for Digital Times."

Roberts, Charlie, and Graham Wakefield. 2018. "Tensions and Techniques in Live Coding Performance."

Rohrhuber, Julian, and Alberto de Campo. 2004. "Waiting and Uncertainty in Computer Music Networks." In *ICMC*.

Snyder, Bob. 2000. *Music and Memory: An Introduction*. MIT press.

Tanimoto, Steven L. 1990. "VIVA: A Visual Language for Image Processing." *Journal of Visual Languages & Computing* 1 (2): 127–39.

———. 2013. "A Perspective on the Evolution of Live Programming." In *2013 1st International Workshop on Live Programming (LIVE)*, 31–34. IEEE.

Wilson, E, S Lawson, A McLean, J Stewart, et al. 2021. "Autonomous Creation of Musical Pattern from Types and Models in Live Coding." *xCoAx 2021 9th Conference on Computation, Communication, Aesthetics & X*.

Wilson, Elizabeth, Gyorgy Fazekas, and Geraint Wiggins. 2023. "On the Integration of Machine Agents into Live Coding." *Organised Sound* 28 (2): 305–14. https://doi.org/10.1017/S1355771823000420.