An analytical framework for musical live coding systems based on gestural interactions in performance practices

Proceedings of the International Conference on Live Coding, Valdivia, December 2021
https://iclc.toplap.org/2021/

An analytical framework for musical live coding systems based on gestural interactions in performance practices

Georgios Diapoulis
Interaction Design, Department of Computer Science & Engineering, Chalmers University of Technology, University of Gothenburg
geodia@chalmers.se

Palle Dahlstedt
Interaction Design, Department of Computer Science & Engineering, Chalmers University of Technology, University of Gothenburg
palle@chalmers.se

ABSTRACT

Gestural interaction in live coding performance is still in its infancy, albeit the long tradition in music performance studies. Computational challenges in musical live coding have been motivating the research community towards the development of novel programming languages and interfaces. On the other hand, given the maturity of many music systems there is an increasing demand for theory building on live coding systems and practices. Here, we present an observational study from videos of live performances available online and we introduce an analytical framework for live coding music systems. We begin by examining how performance practices differ on potentially equivalent systems. On the spotlight of the framework is the viewpoint of gestural interactions under the prism of music psychology and perception. We examined several systems on three main processes: (i) interface design, (ii) gestural mapping and (iii) user's interaction. These processes are presented as an orthogonal three-dimensional framework, so to facilitate visualizations and readers' understanding. Preliminary assessments of the systems in question agree with ground truth knowledge of the computational classification of the systems. Furthermore, we analyze a few notable systems that are stretching the boundaries of our dimensional framework, indicating that more dimensions may be required. Finally, we discuss the analytical framework in relation to a higher-level description of live coding music performance and we discuss future studies that may be conducted to assess the validity of this approach.

1. INTRODUCTION

Gestures are an integral part in our daily interactions with computing technologies. There has been increasing interest in mobile devices that afford gestural interactions, typically through touchscreen displays, which in return are progressively transforming user's interactions. In psychology of programming, users are seen as experts (Blackwell & Collins, 2005) and users' practices may influence technological developments. In analogy to this liberated stance of the division between users versus experts, here, we take a liberated stance on live coding. We adhere any declarations of live coding practices as live coding, without a need for strict definitions (Collins, 2011). In the context of music performance, gestures have been studied extensively in both music psychology and music perception. There is a broad consensus that musical gestures carry functional, aesthetical and social aspects (Godøy & Leman, 2010).

1.1 Gestural interaction in live coding research

In live coding, gestural interactions are seen as an indispensable part which can be improved in terms of virtuosity and expressivity by extensive practice (Collins, 2011). Yet, since live coding as performance practice is still in its infancy, there are no methods on how to master gestural control in performance. Many practitioners have reported embodiment during live coding performance. More specifically, Baalman (2015) reflects on typing automaticity that is developed through the familiarity with a certain programming language. Given such anecdotal evidence, action execution is linked to the mental model that we form by extensive practice with the programming language of our preference.

In the context of traditional music performance, action execution is linked to imagery of auditory percepts (Keller & Koch, 2008). Such, imagery percepts are fed to gestural unfoldings as these are realized by sequences of musical gestures during performance. Auditory and motor perception are the driving forces of these realizations and both effortful and involuntary imagery may be contributing factors during music making in live coding (Diapoulis & Dahlstedt, 2021). Given the extensive study and experimental evidence on auditory and motoric skills during performance, musical interface design studies have been taking advantage and building numerous evaluation frameworks for music systems. Gestural control in computer music interfaces has already a long tradition and there are numerous frameworks and evaluation studies. Indicatively, in the NIME community[1] a meta-analysis revealed that within the years 2012-2014 more that 200 studies related to evaluation had been reported (Barbosa et al., 2015).

Evaluation systems in musical interaction design have been proposed since the 90s, and in 00s there is already a steady ground which has a parallel coevolution with trends in human-computer interaction (Wanderley & Orio, 2002). Early studies on evaluation of musical interfaces have been focusing on assessments of simple interactions. Later, developments of mixed methods were employed to evaluate authenticity of artificial agents (Stowell et al., 2009). Despite all these efforts, it has been noted that most systems are used by a single performer, who is typically the developer of the system (Barbosa et al., 2015). This makes difficult to build a solid background on evaluation methods of digital musical instruments (DMIs). In the same study were identified some basic components among the reviewed articles, which are related to the *criteria* of the evaluation, the *methods* used and the *goals* of the evaluations among others.

2. CRITERIA, GOALS AND METHODS FOR AN ANALYTICAL FRAMEWORK

Here, we will present the criteria, goals and methods of our study. The main *goal* is to present a preliminary analytical framework for live coding music systems. Our starting point are the gestural interactions with the musical interface. The *method* is based on an observation study of videos available online (see Table 1). The main *criteria* are (i) to identify performance practices that show broad variations on the gestural interactions, and (ii) to examine how performance practices may differ in potentially equivalent systems. In that manner, we did not include a broad variety of "standard" or "canonical" (Roberts and Wakefield, 2018) live coding systems, because they demonstrate minimal variation in gestural interactions. By "standard live coding" systems, we address all practices that are using the keyboard as the main input interface and a typical programming language. Here, a typical programming language, refers to any programming language which can make use of an interpreter or a compiler, and requires typed programming expressions that are well-formed in a text editor. Instead, in this study we focused on highly individualistic systems that build upon anything from low-level computing interfaces to high-level systems. These may include from traditional musical instruments, like the piano, to printed circuit boards and hardware prototypes on solderless breadboards.

	Author	System	Video
1	Baalman	Code LiveCode Live	https://vimeo.com/434679284
2	Collins	Type-A personality	https://youtu.be/0fX0AymCtgA
3	Diapoulis	stateLogic machine	https://vimeo.com/43121821

[1]New Interfaces for Musical Expression, https://www.nime.org/.

4	Griffiths	Al-jazzari	https://youtu.be/Uve4qStSJq4
5	Magnusson	Threnoscope	https://vimeo.com/63335988
6	McLean	TidalCycles	https://youtu.be/PeyE8ATMezs
7	Noriega & Veinberg	CodeKlavier CKalkulator	https://youtu.be/hD-PWNDebD4
8	Noriega & Veinberg	CodeKlavier hello world	https://youtu.be/ytpB8FB6VTU
9	Reus	iMac Music	https://vimeo.com/205714278
10	Salazar	Auraglyph	https://youtu.be/qqt2vSNy_nA

Table 1. Video material for the observational study.

2.1 Criteria

The key criterion for the analytical framework is the gestural interaction with the musical interface. More specifically, we aim to identify whether the gestural interactions have any impact on the running algorithm of the system. Given the anecdotal evidence that a programming language may influence action (motor) execution (Baalman, 2015), it is reasonable to think how execution of gestural interactions can change our mental model of the running algorithm. Also, we have identified that there is increasing interest in cases where performance practices show variations on potentially equivalent systems (Diapoulis & Dahlstedt, 2021). Furthermore, to ease the theoretical analysis we excluded any visual percepts and we focused only on auditory and motor perception.

2.2 Goals

Our goal is to present a preliminary analytical framework for musical live coding systems. We coupled this framework to a theoretical background which may account for a high-level description of live coding music performance. Furthermore, we aim to systematize the study of gestural interactions in live coding performance, which may support the development of future experimental studies in the psychology and perception of live coding.

2.3 Methods

The main method was that we coded videos of live coding performances, based on subjective evaluations of the first author (see Table 3). Having in mind the key criterion of gestural interactions, we attempt to identify which might be the most important factors that differentiate these performance systems. Also, we constrained these factors up to three to facilitate visual communication and reader's understanding. Our method differs from previous studies in the live coding community. These have been ranging from aesthetic evaluation studies (Bell, 2013), to position articles on interface design (Stowell & McLean, 2013) and theoretical approaches on musical gestures (Jarvis, 2019) and cognitive processes (Sayer, 2015). Here, we present an alternative view based on analyzing videos online and we attempt to bridge studies from music psychology and perception within live coding research (Tanimoto, 2017). Our view stems from embodied music cognition (Leman, 2008) which moves beyond a view of input-output processes of human perception and cognition, and on previous work on embodied playing with algorithms (Dahlstedt, 2018). On this background and given a subjective perspective of the first author on live coding practices we are presenting a preliminary analytical framework of musical live coding systems. From the observed video material that is available online (see Table 1) we extracted meaningful abstractions and delivered a preliminary framework that can be discussed, challenged modified and extended by the live coding community. While we also discuss on live coding practices and agents the focus on this study is on live coding systems.

3. SETTING THE GROUND FOR AN ANALYTICAL FRAMEWORK

In this section, we start with a high-level description of live coding music performance. Then, we present our view on how music psychology and music perception may be related to live coding practice. Following that, we try to link this view to music cognition and agency in live coding. Finally, in the next section we introduce a preliminary dimensional framework for musical live coding systems.

We see that a three-fold description of practices, agents and systems is at the very center of studies in live coding music performance (see Figure 1, bottom row). The human companion is an indispensable part in live coding practice (Collins et al., 2003). This is because even if the musical agents may be fully autonomous still the training corpus is based on code written by humans (Stewart and Lawson, 2019). Although we may imagine fully-autonomous systems which may not be based on humanly-written code for the training data, yet, some amount of human agency is being transferred from the very foundations of computer science (Dahlstedt, 2021). Also, aspects of human-machine musicianship that arise during a live coding performance are shifting the boundaries of agency in music performance (Brown, 2016), and sophisticated designs may offer symbionts of human-machine musicianship (Collins, 2015). Finally, a music system seems to be a necessary component for musical live coding performances, regardless of seeing this as a musical notation system or a formal computational language (Magnusson, 2011).
Music systems show a broad diversity from low level computing components to high level languages and interface setups. Consequently, practices also show a broad diversity which is influenced to some extent by the design decisions of the systems. Finally, agents within the context of live coding music performance also show a broad diversity, ranging from human agents, either expert/novice programmers or expert/novice musicians, to artificial autonomous agents based on machine learning and machine listening. Indicatively, Nick Collins presented two alternatives, which may also overlap to some extent, as machine listening control of live coding, or live coding control of machine listening (Collins, 2015).

3.1 Musical activities in live coding practice

We start our high-level description of live coding performance by discussing how musical activities may be experienced in live coding practice. Musical activities can be divided to three overlapping categories, music making, music listening and musical imagery (Luck, 2015). We may also claim that there is a progressive level of engagement within musical activities, starting from least engagement in musical imagery, more engagement into music listening and even more in music making (see Figure 1, Activity). Music making involves both music composition and music performance. Skilled musical performance is achieved when the musician is exposed to repetitive practice of an activity. The same applies to live coding practice (Nilson, 2007).

3.2 Music perception in live coding performance

3.2.1 Music listening and appreciation

Music listening is the most wide-spread musical activity. We are exposed many hours per day to music, even involuntary. During music performance, either traditional or computer music performance, the musician is both making and listening to the generated sounds and appreciates online music percepts. Contrary to traditional music performance, in computer music the generation of sounds is cloaked within circuits and other high-level components. Thus, in the absence of any visual and haptic feedback (which is typically predominant in traditional music performance), in computer music we are exclusively relying in our auditory capabilities. This is done by taking advantage of our ability to segment sound events with audition. While onset and offset detection can be a notoriously difficult task for machine listening, and may be a philosophical enquiry on its own (Toiviainen, 2015; personal communication), in computer generated music there is no alternative, other than to embody sound events just by listening to them (Palmer, 1997). As a result, here, we excluded any visual percepts during musical live coding. This is because the visual channel is a another complicated mechanism and during live coding it can have a very important contribution to the experience. This decision to focus on motor and auditory percepts may also contribute to the discussion of live coding for blind and visually impaired people (Vetter, 2020).

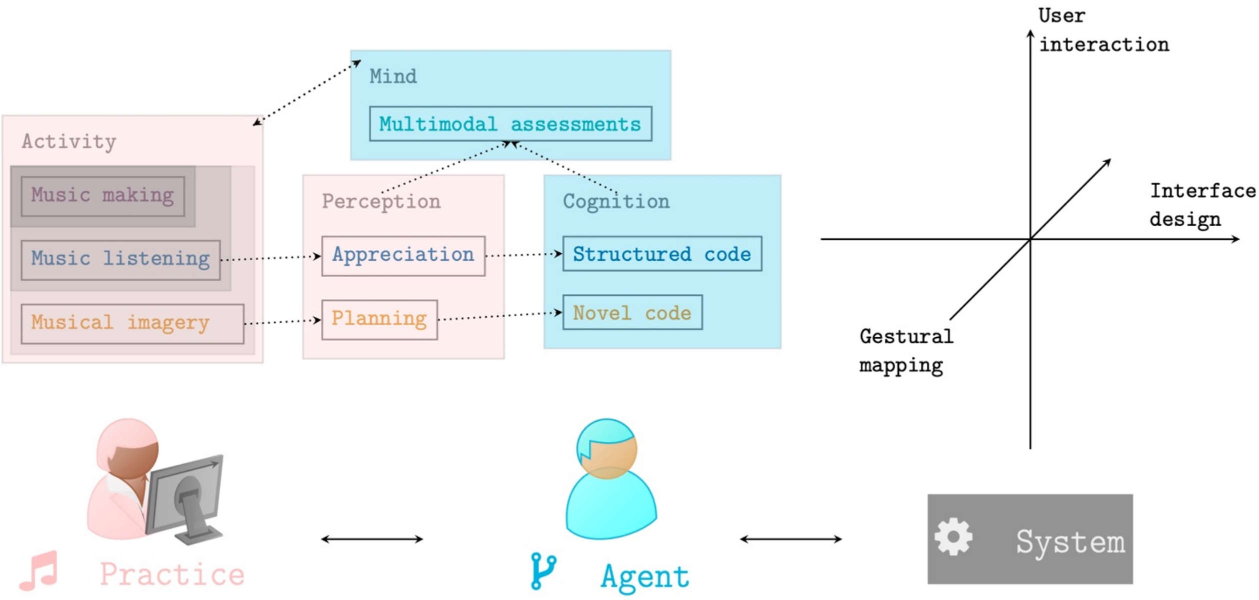

Figure 1. High-level description of live coding music performance: practices, agents, systems; including the subcomponents of musical activities, music perception and cognition in performance and the dimensional analytical framework of live coding systems.

3.2.2 Musical imagery and planning

Musical imagery is when a melody of a song comes to our mind. The so-called earworms, exemplify the phenomenon of involuntary musical imagery. During music performance, we employ both involuntary and effortful music percepts (Keller, 2012). Interestingly, involuntary imagery can also trigger motor execution (Keller & Koch, 2008). Thus, a blend of effortful and involuntary imageries take place during performance. Musical imagery is intertwined to anticipated music percepts and Godøy (2003) has suggested that gestural imagery is an integral part in music performance. He elaborates on that and hints that our mental capacity enable us to compress gestural unfoldings in time, while the same is not true for auditory percepts. One cannot compress a musical sound and experience the same qualities. In live coding we are planning our future actions by trial-and-error, which is an act of novelty (Tanimoto, 2017). On this account, auditory percepts can trigger gestural unfoldings during a live coding performance, which in return may contribute to planning of abstract actions (Diapoulis & Dahlstedt, 2021). Interestingly, musical notation can also trigger involuntary imagery, which is known as notational audiation in music literature (Keller, 2012).

3.3 Music cognition and code structures

As presented above, musical planning is a combination of gestural and auditory anticipatory percepts and music listening can be employed to experience segmentations of the generated sounds. Also, when we appreciate a music percept, this may result to structured code in live coding practice (see Figure 1, Cognition). The question arises how such sounds may be used to change our mental model of the running program? Here, sound segmentation is seen as the only informative unit which we can employ to modify and structure our code. In fact, algorithms are not structures which afford segmentation themselves. Furthermore, gestures also do not exhibit 'well formed' characteristics and it can be difficult even for human annotators to segment gestural unfoldings. Here, we see that segmentations that are formed from auditory percepts can inform gestural interactions in planning future action executions. In that manner, planning contributes to novel code evaluations (Diapoulis & Dahlstedt, 2021).

3.4 How agency appears in live coding

Artificial agency is an immediate consequence during live coding practice (Brown, 2016). This is further entangled when machine learning or machine listening components are involved during the activity of music making (Collins, 2015). For instance, when a machine listening component is being involved, then we can think of this as an augmentation to our hearing. When we control code structures or parameters with machine listening processes then the running program can be thought as an agent that applies semantic (code) adjustments that are driven by another modality. In that manner, the electronics are producing mechanical energy which vibrates air molecules and surrounding structures and then is processed again through digital logic to apply adjustments on its own structure. In fact, this can be also done without rendering the physical sounds, but we exemplify the physical process to facilitate reader's understanding. These practices are shifting conventional agencies in musicianship. In traditional music performance the musicians can embody expressive intentions in a clear manner and the visual channel biases our perception of expressivity (Davidson, 1993). How may we study expressivity in such symbionts of human-machine musicianship?

4. LIVE CODING SYSTEMS

A live coding system is a rather complicated structure. All the fruitful efforts of the live coding community to deliver systems that can enable live performances and even algorave parties has come to a rather matured state, in comparison to 10 years ago. Some of the most prominent systems are, for example, ixi-lang (Magnusson, 2011), SonicPi (Aaron, 2016) and TidalCycles (McLean & Wiggins, 2010). A common feature of all abovementioned systems is that they are based on elegant code expressions which foster immediacy during performance and may also ease educational purposes. Indicatively, ixi-lang was developed with a constraint of 5 seconds per command so to facilitate live performance. The technological demands of such developments and the fact that the original authors had to develop most of the system on their own, may have hindered other aspects of these developments. For instance, from the viewpoint of gestural interactivity these systems demonstrate equivalence at first glance. This is because, the composer-programmer is typing programming commands on a keyboard, which includes algorithmic complexity, and waiting until she has an executable command ready to be successfully evaluated by the interpreter. Here, we refer to this category of systems as “standard live coding” systems.

4.1 A three-dimensional analytical framework for musical live coding systems

In this section, we present a three-dimensional analytical framework (Diapoulis & Dahlstedt, 2021), equipped with a dyadic condition, here called code-first and music-first, as proposed by Tanimoto (2017). Each dimension represents a process as shown in Table 2, and is equipped with two semantic differential concepts. On the lower end is a low-level concept, also called concrete, and on the upper end a high-level concept, also called abstract. For the dyadic condition we assigned code-first as a low-level concept in analogy to how a musical score affords different interpretations during music performance. Algorithms are seen as scores in live coding (Magnusson, 2011), and in this case code is seen as more concrete in comparison to the generated music which may differ for example in different music halls, sound reproduction systems and the like.

	Process	Low-level (concrete)	High-level (abstract)
X axis	Interface design (ID)	Literal design	Metaphorical design
Y axis	Gestural mapping (GM)	Algorithmic significance	Algorithm agnostic
Z axis	User interaction (UI)	Direct manipulation	Algorithmic complexity
Dyadic condition	Sound generation (SG)	Code-first	Music-first

Table 2. Dimensional analytical framework for musical live coding systems.

The first dimension, interface design, refers to the concept of how literal or metaphorical is the design of the interface. By literal design we refer to any design decisions that rely on conventional programming interfaces, like a text editor or a hardware prototype equipped with printed circuit boards, switches, buttons and the like. Metaphorical design refers to any design decisions which may conceal the programming activity, like playing the piano or playing a video game. The second dimension on gestural mapping examines what is the effect of gestural interactions on the running algorithm. For instance, during a “standard live coding” the musician is typing on the keyboard without any temporal or other constrains. In that manner, the live coder is doing as many gestures as she likes until the code execution is successful. We call this process algorithm agnostic. On the other hand, if the gestural unfoldings are modifying the structure of the running algorithm we call this process algorithmic significant. The best example to understand this dimension is to watch the performances by Noriega & Veinberg “hello world” and “CKalkuλator”. In these two different setups of the CodeKlavier system, the pianist is typing on the musical keyboard (see “hello world”), or just playing the piano (see “CKalkuλator”). We see that in the “hello world” performance the gestural mapping is agnostic to the algorithm, while in “CKalkuλator” the gestures are modifying the running algorithm. Finally, the third dimension shows the semantic differentials of direct manipulation and algorithmic complexity. Defining direct manipulation in the context of live coding can be a challenging endeavour. We see that a musical interface which facilitates recognition instead of retrieval, may be classified as exhibiting direct manipulation.

Table 3 below shows a color coding for the systems examined in this study. The uppercase “L” stands for low-level concepts and the uppercase “H” for high-level concepts. When the systems in question afford both low-level and high-level concepts, we coded such cases as “L/H”.

	Author	System	ID	GM	UI	SG
1	Baalman	Code LiveCode Live	L	H	H	H
2	Collins	Type-A personality	L/H	L/H	L	H
3	Diapoulis	stateLogic machine	L	L	L	L
4	Griffiths	Al-jazzari	H	L	L	L
5	Magnusson	Threnoscope	L/H	H	L/H	L
6	McLean	TidalCycles	L	H	H	L
7	Noriega & Veinberg	CodeKlavier CKalkuλator	H	L	H	H
8	Noriega & Veinberg	CodeKlavier hello world	H	H	H	H
9	Reus	iMac Music	L	L	L	H

10	Salazar	Auraglyph	H	H	H	L

Table 3. "L" for low-level concepts and "H" for high-level concepts. "ID": interface design, "GM": gestural mapping, "UI": user interaction, "SG": sound generation.

5. VISUAL REPRESENTATION OF LIVE CODING SYSTEMS IN A DIMENSIONAL FRAMEWORK

Figure 2 shows a three-dimensional representation of the analytical framework. This is based on an orthogonal coordinate system, which may be misleading for the reader as orthogonality typically refers to independent concepts. It is interesting to observe that in this spatial representation Marije's Baalman "Code LiveCode Live" system overlaps to a "standard live coding" system (see Table 3, McLean). Also, the "stateLogic machine" by Diapoulis overlaps with "iMac Music" by Reus. Finally, the performance "hello world" overlaps with "Auraglyph" system. Interestingly, in all these pairs the systems differ on the dyadic condition code-first and music-first. From the observation that the "Code LiveCode Live" differs from a "standard live coding" system only in the code-first condition, Marije's system assigned meaning to the act of typing on the keyboard (Diapoulis & Dahlstedt, 2021). What more can we learn from these overlapping systems that differ only on the dyadic condition code-first, music-first?

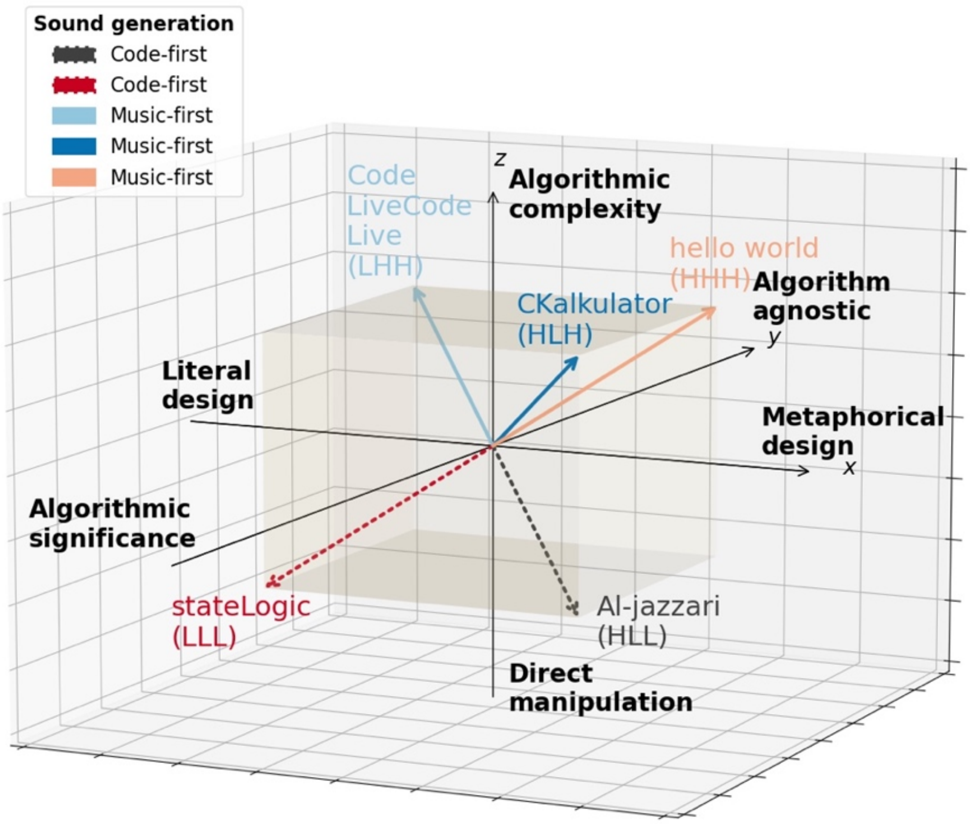

Figure 2. Three-dimensional analytical framework for musical live coding systems. X: interface design (ID), Y: gestural mapping (GM), Z: user interaction (UI). The dyadic condition on sound generation (SG) is shown with dashed arrows.

Based on Table 3 and Figure 2 we should note that a preliminary validation of the systems in question is provided by the directionality of the cognitive design that we selected for the semantic differentials. The classification as low-level and high-level on each system agrees to ground truth knowledge of the workings of the systems. For instance, “Al-jazzari” is indeed based on low-level computing processes and is classified as such in all dimensions except the interface design (ID) axis.

6. DISCUSSION

We have introduced a preliminary three-dimensional analytical framework for musical live coding systems. Here, we attempt to bridge the gap between studies in music psychology and perception and to shift perceptions on fundamental differences between traditional and live coding music performance (Sayer, 2015; Tanimoto, 2017). The focal point of the study was to examine live coding systems from the viewpoint of gestural interactions. That was a revealing point of departure as we identified how potentially equivalent systems can bring about meaning to gestural interactions in live coding performance (Diapoulis & Dahlstedt, 2021).

We coded videos from performance practices available online, based on subjective evaluations of the first author. We identified early on, based on personal experience, that metaphorical design is particularly important within the broad diversity of live coding systems. Such systems can be quite surprising for someone somehow familiar with live coding, when she attends a live coding performance in which the performer seems to be engaged in a video game activity. Furthermore, based on personal reflections when designing a live coding prototype, it is reasonable to experience the dominance of the code-first requirement. When designing his “stateLogic machine”, the first author (GD) tried to minimize this notably anxious waiting time, but he realized that you must always wait for the next positive edge clock. At least we cannot see any other way except if we move to a different computational paradigm that the input information is read on-demand and not periodically in time. About the second dimension on gestural mapping, we reflected on the literature of musical gestures in music performance (Godøy & Leman, 2010). In analogy to the primary and secondary aspects of musical gestures, we thought how musical gestures in live coding can be significant to the running algorithm. In traditional music performance primary gestures refer to sound-producing gestures, whereas the secondary gestures typically may carry emotions or facilitate communication. During live coding, the sound-producing part cannot correspond to the traditional meaning of sound-producing gestures, as the sound generation is performed using digital signal processing algorithms. Here, we identified that musical gestures in live coding can either have immediate impact on the running algorithm or can be ignorant about the workings of the algorithm. For instance, in Baalman’s performance the musical gestures are sound-producing gestures, but they do not change the structure of the running algorithm. This because the modifications are performed on the parameter level. Contrary, in CKalkuλator the pianist is programming by playing the piano. Here, there is a fine line between what we adhere as programming or not. For example, in Threnoscope, Thor Magnusson is applying direct manipulation with the mouse to modified parameters of the running algorithm. We also classified this interaction as algorithm agnostic, due to the fact it does not apply changes to the structure of the running algorithm. A structural change on the running algorithm, is exemplified by Kiefer’s (2015) “approximate programming”. Specifically, this is exemplified by the real-time visualizations of the synths structure as these are shown using hierarchical trees[2]. An important note is that in computer music an algorithm is expressed using binary digits. One may question how a gesture may have an effect on a one-dimensional structure (Collins, 2016). Here, we see an algorithm as an abstraction that corresponds to a mental model and may be influence by embodiment in performance (Fanfani et al., 2020). Finally, the third dimension on user interaction was employed by studies in psychology of programming and human-computer interaction (Diapoulis & Dahlstedt, 2021). Sometimes, it can be difficult to evaluate whether a musical interface affords direct manipulation or algorithmic complexity. It should be noted that interfaces which support recognition processes are seen as exhibitors of direct manipulation.

[2] https://github.com/chriskiefer/ApproximateProgramming_SC/blob/master/approxTree.scd

On the dyadic condition code-first/music-first we would like to make an analogy to traditional musical instruments. Let us think for a moment how musical instruments like the qanun or the organ are changing music systems during performance. In both instruments the musician apply on-the-fly changes to the system. The qanun has a mandal technology which enables the performer to adjust the length of the strings. In principle, the mandal technology is changing the melodic modes, also known as maqams, that the instrument affords. Is this mandal technology a precursor to changing the program as it is running? Is this form of interaction a precursor of the dyadic condition code-first and music-first? Certainly, moving a mandal on the qanun is not a sound-producing gesture, but a necessary one.

In our study there is a predominant absence of collaborative live coding systems and practices. We did not examine this category of systems intentionally, due to the broad new perspective that collaborative live coding brings about in the development of the field. Performing with other people is seen as one of the most difficult tasks. If we would like to represent collaborative performance systems with the approach presented here, it is reasonable to assume that more dimensions are required for a more inclusive framework. A straight-forward solution to this can be a multi-dimensional space based on design space analysis (Birnbaum et al., 2005).

An important result from the present study is that a preliminary validation is provided by the computational classification of the systems, which agrees to ground truth knowledge. For instance, systems that are based on low-level computing processes are also classified as low-level systems in our dimensional framework. This is because we followed a cognitive paradigm which assigns the directionality of the dimensions from low-level to high-level concepts.

Finally, future studies should validate the framework and propose new dimensions for a more inclusive framework. Such frameworks may be used by both practitioners and academics for either creative explorations or theoretical discussion and design of experimental studies. This direction will contribute to the psychology and perception of live coding. For instance, interview studies with the original authors, or other live coders, may be employed to verify shared conceptions among the community. Also, questionnaire studies may validate, or not, the semantic differentials. If such efforts proved successful, then we may have to start thinking how we may browse such music spaces during live coding practices (ie. Type-A personality and Threnoscope). Creative exploration of such dimensional spaces should be seen as a helpful tool and as a challenge to move beyond the expressive capacity of such developments.

Acknowledgments

Special thanks to Mafalda Samuelsson-Gamboa for reviewing and Kivanç Tatar for input on this article.

REFERENCES

Aaron, Sam. "Sonic Pi–performance in education, technology and art." International Journal of Performance Arts and Digital Media 12, no. 2 (2016): 171-178.

Baalman, Marije. "Embodiment of code." In Proceedings of the First International Conference on Live Coding, pp. 35-40. 2015.

Barbosa, Jeronimo, Joseph Malloch, Marcelo M. Wanderley, and Stéphane Huot. "What does" Evaluation" mean for the NIME community?." (2015).

Bell, Renick. "Towards useful aesthetic evaluations of live coding." In ICMC. 2013.

Birnbaum, David, Rebecca Fiebrink, Joseph Malloch, and Marcelo M. Wanderley. "Towards a dimension space for musical devices." In Proceedings of the 2005 conference on New interfaces for musical expression, pp. 192-195. 2005.

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

Brown, Andrew R. "Performing with the other: the relationship of musician and machine in live coding." International Journal of Performance Arts and Digital Media 12, no. 2 (2016): 179-186.

Collins, Nicolas. "Semiconducting: Making music after the transistor." Musical Listening in the Age of Technological Reproduction. Routledge, 2016. 313-322.

Collins, Nick, et al. "Live coding in laptop performance." Organised sound 8.3 (2003): 321-330.

Collins, Nick. "Live coding of consequence." Leonardo 44, no. 3 (2011): 207-211.

Collins, Nick. "Live Coding and Machine Listening." In Proceedings of the International Conference on Live Coding, pp. 4-11. 2015.

Dahlstedt, Palle. "Action and Perception: Embodying Algorithms and the Extended Mind." The Oxford Handbook of Algorithmic Music, Oxford University Press, 2018. 41-65.

Dahlstedt, Palle. "Musicking with Algorithms: Thoughts on Artificial Intelligence, Creativity, and Agency." Handbook of Artificial Intelligence for Music. Springer, Cham, 2021. 873-914.

Davidson, Jane W. "Visual perception of performance manner in the movements of solo musicians." Psychology of music 21, no. 2 (1993): 103-113.

Diapoulis, Georgios, and Palle Dahlstedt. (2021). "The creative act of live coding practice in music performance." In PPIG 2021 Doctoral Consortium, University of York, York, UK.

Fanfani, Giovanni, et al. "(Micro-) Performing Ancient Weaving in the PENELOPE project." Performance Research 25.3 (2020): 123-130.

Godøy, Rolf Inge. "Gestural imagery in the service of musical imagery." In International Gesture Workshop, pp. 55-62. Springer, Berlin, Heidelberg, 2003.

Godøy, Rolf Inge, and Marc Leman, eds. Musical gestures: Sound, movement, and meaning. Routledge, 2010.

Jarvis, Ian. "Live coding: sound - gesture - algorithm." In Proceedings of the international conference on live coding. Prado, Madrid (2019).

Keller, Peter E. "Mental imagery in music performance: underlying mechanisms and potential benefits." Annals of the New York Academy of Sciences 1252, no. 1 (2012): 206-213.

Keller, Peter E., and Iring Koch. "Action planning in sequential skills: Relations to music performance." Quarterly Journal of Experimental Psychology 61, no. 2 (2008): 275-291.

Kiefer, Chris. "Approximate Programming: Coding through Gesture and Numerical Processes." Proceedings of the First International Conference on Live Coding, ICSRiM, University of Leeds. 2015.

Leman, Marc. Embodied music cognition and mediation technology. MIT press, 2008.

Luck, Geoff. "Lecture notes in the x-factor in music." University of Jyväskylä, Jyväskylä, Finland (2015).

Magnusson, Thor. "Algorithms as scores: Coding live music." Leonardo Music Journal 21 (2011): 19-23.

McLean, Alex, and Geraint Wiggins. "Tidal–pattern language for the live coding of music." In Proceedings of the 7th sound and music computing conference. 2010.

Nilson, Click. "Live coding practice." Proceedings of the 7th international conference on New interfaces for musical expression. 2007.

Palmer, Caroline. "Music performance." Annual review of psychology 48, no. 1 (1997): 115-138.

Roberts, Charlie, and Graham Wakefield. "Tensions and Techniques in Live Coding Performance." (2018): 293-317.

Sayer, Timothy. "Cognition and improvisation: some implications for live coding." (2015).

Stewart, Jeremy, and Shawn Lawson. "Cibo: An Autonomous TidalCyles Performer." Proceedings of the Fourth International Conference on Live Coding. 2019.

Stowell, Dan, and Alex McLean. "Live music-making: A rich open task requires a rich open interface." In Music and human-computer interaction, pp. 139-152. Springer, London, 2013.

Stowell, Dan, Andrew Robertson, Nick Bryan-Kinns, and Mark D. Plumbley. "Evaluation of live human–computer music-making: Quantitative and qualitative approaches." International journal of human-computer studies 67, no. 11 (2009): 960-975.

Tanimoto, Steve. "Challenges for livecoding via acoustic pianos." In 3rd International Conference on Live Coding. Morelia, Mexico. 2017.

Toiviainen, P. Personal communication (2015).

Wanderley, Marcelo Mortensen, and Nicola Orio. "Evaluation of input devices for musical expression: Borrowing tools from hci." Computer Music Journal 26, no. 3 (2002): 62-76.

Vetter, Jens. "WELLE-a web-based music environment for the blind." Proceedings of the International Conference on New Interfaces for Musical Expression. Birmingham, United Kingdom. 2020.