

Interplay Between Composition, Instrument Design and Performance

M.A.J. Baalman

Abstract With electronics and code as an essential part of new musical instruments, the boundaries between composition, instrument design and performance are blurring. With code that can be changed and compiled on the fly, the design of an instrument becomes a fluid process, which can even be a performance in itself. Starting with an example from my own artistic practice, I explore the concepts of composition, instrument and performance and what role the design of electronics and software plays in these. What influences design decisions when developing instrument? How does the materiality of electronics and code inform these decisions? How do the knowledge and skills of the makers play their role in this?

1 Introduction

With electronic and digital media as key elements in musical instrument design, the boundaries between composition, instrument design and performance are blurring. Practitioners do not need to make a clear distinction while developing their projects. However, when they present their work in the context of the established cultural scene, they need to deal with the common division of roles. This paper sheds light on the work process of such artists and how their practice plays in the field between those roles. The questions raised are motivated by my own practice as an artist.

In the following section I will describe my piece “Wezen—Gewording” to motivate the questions I raise in Sect. 3. In Sect. 4 I will look at the discourse about composition, instrument and performing in the context of electronic and computer music. In Sect. 5 I will look at live algorithms or interactive music systems, music software and tools and livecoding. I will conclude with a discussion reflecting back on the questions I pose in the beginning.

M.A.J. Baalman (✉)
Nescivi, Amsterdam, The Netherlands
e-mail: marije@nescivi.nl

2 Becoming

Since August 2013, I have been developing the piece “Wezen—Gewording”.¹ Within this piece, I am looking for a connection between the sonic gesture and the theatrical gesture. I explore this connection by exposing my method of **creating the instrument during the performance, while at the same time playing it** Fig. 1.

Over the course of two-and-a-half years, I performed the piece for 18 times and, both before and during each show, I developed it further. While performing, I go back and forth between body movements and livecoding the mapping between sensor data and its effect on the sound.

I wear gestural controllers at my wrists and hands with a number of buttons in each palm. The controls are designed such that they keep my fingers and hands free enough for me to type on my laptop’s keyboard. These controllers started out as an assembly of elastic bands, perfboards, buttons and a Sense/Stage MiniBee (Baalman 2016). **Over the course of 1.5 years of performing the piece, the design settled in** that I figured out how many buttons I needed, where to place them, and how to mount them onto a wearable, an open glove, that now contains the buttons, the microcontroller and the battery (see Fig. 2).

The prospect of a performance acted for me as a kind of *showtime-deadline*, before which I did a series of rehearsals and spent time developing the code and hardware. The performance itself had a duration of 10–15 min.

Code development meant to create a framework, within which I can quickly create connections between gestures and sound during performance. Both rehearsals and performances informed me whether my design decisions worked, whether I needed to revert them, or in which direction I had to further develop them. The mapping and sound synthesis algorithms that I wrote while playing were sorted out during the design and development time, based on whether I found they gave interesting results during the performance. Thus **I assembled a collection of code snippets for use in future performances**.

Furthermore, I wrote and rewrote mapping architecture, allowing for shorter ways to implement the same kind of behaviour during performance, **abstracting concepts I worked out in previous performances. Another part of the work was reverting these abstractions**, as I discovered during playing that they did in fact impede the improvisation, rather than support it.

Over time, I added new and substantial elements like interactive lights or additional sensors on various parts of the body to the performance. At the same time, I found it necessary to add less direct interaction layers, which resulted in the inclusion of a gesture recognition algorithm, triggering subtle changes in the background soundscape. **Sometimes these additions would become a constant part of the system, at other times they disappeared** after one or more performances.

My role shifted throughout the whole process, from being the actor and mover to being programmer, from electrical engineer to garment maker. This means that,

¹“Wezen—Gewording” can roughly be translated to *Being—Becoming*.



Fig. 1 The performance “Wezen—Gewording” in May 2015 at the Kunst Achter Dijken festival in Pingjum, The Netherlands

since I embody movements and listen to many variations of their sonification, I am aware of the kind of data that particular movements will create. I can incorporate this knowledge into programming a sound and its mapping to the sensors. At the same time, I create new scenarios by code in which I can explore my movements and their effect on the sound. The sound informs my movement, which in turn informs me in my subsequent mapping decisions.

While rehearsing for a particular performance I develop a rough structure. I come up with different directions for the piece to develop, and if these were hard to reach before, I try to make them accessible by adapting the code framework. During the livecoding within the performance, I am aware of the framework I built and in which I make changes. I recall the limits of easily modified code and what possibilities it offers to the me as the mover. Thus, there is a tight connection between my embodied knowledge of moving and listening, and my engineering knowledge of how the technology enabling the performance works.



Fig. 2 The three stages in the development of the gestural controllers of “Wezen—Gewording”. At the *top* the first version, *middle* the second, and *bottom* the final version

3 ‘Where Is the Piece?’

In the process described above, it is hard to determine whether the work I do is composing, building and designing an instrument, or performing. It is even hard to say what “Gewording” is. Did I compose it? Is it a performance? Is it an instrument? Can the different performances be considered the same piece?

John Richards poses similar questions in his article “Lead and Schemas” (Richards 2011) about his work on Dirty Electronics:

‘Where is the piece? Is it in the process of building an instrument, the instrument itself, a notated score, the schematic, or the live performance? Another is, At what point does interpretation come into it?’

For my piece “Wezen—Gewording”, I have a vague understanding of which parts are the composition, the instrument or the performance:

- The *composition* is the concept of combining gesture, sound and (partially) livecoded software, and the (roughly) planned flow of procedure in which I perform with these elements.
- The *instrument* is the combination of the laptop, the software I wrote, the controllers I built, and the livecoding API² I implemented.
- The *performance* is the moment of playing the instrument in front of the audience. I call the preparation for a performance the *rehearsal*.

Yet, those elements are inseparable. My choices in the composition are informed by how I perform and my compositional needs inform how I (re)design the instrument. The instrument design again informs how I (re)do the composition and how I perform. While performing, I embody the possibilities both of the instrument as well as the composition.

The software or code seems to be what blurs the distinctions between the different elements. It is hard to distinguish if a particular segment of code is part of the instrument, of the composition, or even the performance, or perhaps all of these at the same time.

How do other artists relate to their practice in the field of electronic and computer music? What do they consider a composition, an instrument or a performance? What role does code play in these considerations?

4 Composition, Instrument and Performance

4.1 Composition

The basis of the word composition comes from the Latin verb *componere*, putting together (Oxford Dictionaries 2016), also used for ordering or arranging, i.e. putting an order to something. On Wikipedia (2016) musical composition is described as:

Musical composition can refer to an original piece of music, the structure of a musical piece, or the process of creating a new piece of music. People who practice composition are called composers. “Composition” is the act or practice of creating a song or other piece of music.

²API stands for *application programming interface*; it defines which functions can be called from a library. In the context of livecoding it defines the set of commands that can be called during the performance, thus determining the vocabulary that is available.

The article gives a description of the elements that are present in different musical traditions, such as the use of music notation to communicate the composition to performers (who then interpret the composition), orchestration or arrangement for a set of instruments, and how improvisation is allowed or expected from the performers by the composers or by the musical tradition to which the pieces belong.

Maxfield (1963) writes about the contrast between instrumental music and “electronic music for want of a better term”:

Formal instrumental music is generally embodied in written score
the notation of which is not to be taken as complete or exact;
thus room is left its interpreter
to decide nuance of detail
anew for each performance.

But although a recording of an instrumental work
merely projects a *given performance*
in this new art form
it is the composer himself
working directly with the recorded sounds
who selects every nuance;
there may indeed be no score
and later interpretations not desired:
the recording, instead, becomes **the terminal object of creation**.

Thus the composer is the interpreter or performer of his own music, and the music is not distributed by distributing its notation, but a copy of the recording. He further describes how the art object is then fixed, but that “the aesthetic experience it induces is never the same on two different occasions” and that **the rigidity of its structure can only be heard upon second hearing**. This second hearing can be avoided by never listening twice to the same recording of the work, or as he writes: “I frequently compose a new realization for each presentation of a given work.”

While Maxfield (1963) referred to electro-acoustic tape music, for artists building their own electronic circuits to make music, the design of the circuit can be considered as the composition, the electronics as the interpreter, and its schematics as the notation. For example, Lewis (2007) writes:

David Behrman and Gordon Mumma, implicitly advanced the radical idea of a musical composition that could exist purely and entirely in hardware. In this period, scores by the two composers, where they existed at all, often consisted only of a circuit diagram, accompanied by a set of sketchy instructions.

Holzer (2011) writes in his introduction to an issue of *Vague Terrain*:

John Cage once quipped that Serge Tcherepnin’s synthesizer system was ‘the best musical composition that Serge had ever made’, and it is precisely Cage’s reformulation of the concert score from a list of deterministic note values to a set of indeterministic possibilities

that allowed the blurring of lines between instrument-builder and music composer that followed.

Further, he writes about the artists' involvement in the issue:

Their compositions take the form of systems which provide a map of what is possible, but lack a prescribed route on how to get there. The discovery—and the risk—is left to the moment of the performance.

For computer music, one can consider the actual writing of a music program as its composition, and the code as its representation (like a score). **When the code is evaluated, it turns into an interpretation and is performed.**

When looking at the process of composition, it is important to gauge the contributions of all people involved to the final (variant of the) composition. There may be cases where the composer has a concept, which a programmer translates exactly into code (or a circuit) and the code is executed and the composition is interpreted by the machine, unfolding over time, eventually perceived by a listener. However it is more likely that composer and programmer collaborate to fine-tune the code to the composer's intentions based on listening to the outcome. During this process it is likely that the very concept of the composition is adapted. Depending on factors like the skill-set of the programmer, the chosen programming language, or the complexity of the task, **situations may occur in which decisions are made by the programmer that can be interpreted as compositional** and which influence the composer in her conceptual choices. The composition becomes a joint artistic endeavour, hence **single authorship can no longer be claimed**, nor can the role of the programmer be seen as "just engineering". Hayles (2012) writes:

Conceptualization is intimately tied in with implementation, design decisions often have theoretical consequences, algorithms embody reasoning, and navigation carries interpretive weight, so the humanities scholar, graphic designer, and programmer work best when they are in continuous and respectful communication with one another.

Although code may be considered an interpretation of the (idea of a) composition, it is rare that a composer asks several programmers to realize or interpret her composition, or that programmers reinterpret compositions from previous eras. As in the discussion above about tape music: the music is distributed by the code, which set the procedure how a machine will translate the composition to sound. In that sense, the programmer is also the performer.

In other cases, programmer and composer are the same person and roles shift between one and the other or, even more likely, boundaries between them blur into non-existence.

It seems that the act of composition within electronic and computer music is conceptualising and building a system. Both electronic circuit and code can be interpreted either as notation or instrument for a composition. Schnell and Battier (2002) proposed the term *composed instruments* to underline

the fact that computer systems used in musical performance carry as much the notion of an instrument as that of a score, in the sense of determining various aspects of a musical work.

4.2 “New” Instruments and Mapping

In an acoustic instrument the link between the actor in the environment and the resulting output of the medium (sound) is direct or predefined (Miranda and Wanderley 2006). E.g., a performer playing a recorder blows into it and uses her fingers to close holes. These actions, combined with the physical shape of the instrument determine the sound that is heard. The performer adjusts her breath and finger positions trying to align what she hears with what she desires to hear.

Compared to such an acoustic instrument, its electronic equivalent introduces an “arbitrary” factor in its design: material and shape properties no longer determine the sound that can be emitted. Rather, there is a sense of freedom in determining how a gesture creates or modulates a sound. Waisvisz (1999) describes the search for a proper mapping as a very personal process, but that despite this

one can analyse and create distinct relationships between the character changes of a gesture, and the change of musical content—and context—in a way that one’s musical intentions are clearly grasped by listeners.

In the literature and research on the topic of mapping in the context of digital musical instruments (DMI) or new interfaces for musical expression (NIME), the process is mostly seen as and described as a technical area. Waisvisz (1999) on the other hand, believed that

... the algorithm for the translation of sensor data into music control data is a major artistic area; the definition of these relationships is part of the composition of a piece. Here is where one defines the expression field for the performer, which is of great influence on how the piece will be perceived.

Rather than considering only the step of connecting sensor data to synthesis and music control as “mapping” [as Miranda and Wanderley (2006) propose], I consider the full process from physical gesture to (sonic) output “mapping” (Fig. 3), thereby following DelaHunta (2001)’s argument, that mapping constitutes the whole “invisible” part of the instrument. The steps in mapping (Fig. 4) encompass gestures and sensors to capture the gestures, electronic circuits to condition the sensor signals and digitize them, computational models to further condition the data, and eventually coupling of the conditioned data to parameters of the output system. Within the environment, the performer (as well as other spectators) perceives the effect and adjusts her gestures accordingly.

In the design process of an instrument at each of these steps in the mapping, one needs to make decisions. These are ultimately artistic decisions, but they are influenced by external factors, such as available technology, development time, and knowledge and practice of the artists and developers. This process consists of going back and forth between each of the different steps in the mapping process, until finally a satisfactory result is achieved.

Nilsson (2011) describes the design process of his DMI’s in detail and divides the process into “design time” and “play time.” He describes the iterative nature of the design process, alternating between design and play time, while the context of

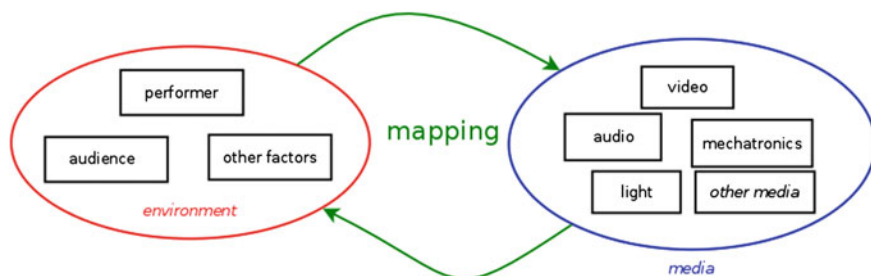


Fig. 3 Mapping between the environment and media—while we directly experience the environment and the media, the mapping is the “invisible” part between the two (DeLaHunta 2001)

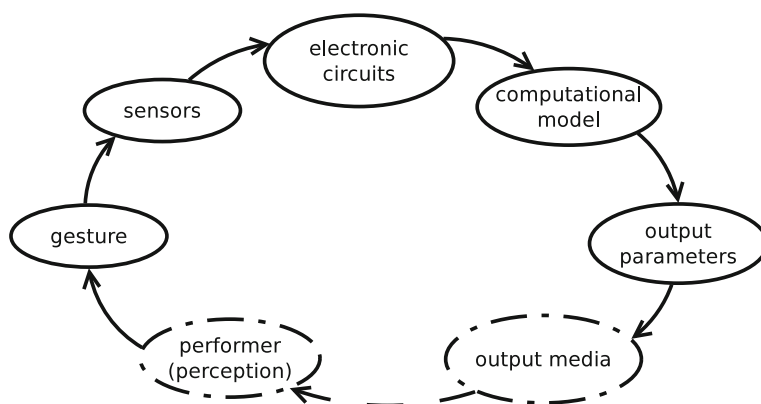


Fig. 4 The different steps involved in mapping from gesture to output media. The steps drawn in a continuous line make up the “invisible” part of the instrument. These are the steps that are adjusted over the course of the building of the instrument

the work grows larger—from private space to the context in which the instrument will be used, to public presentation in concert.

“Design time” is compared to composition, a process taking place “out of time”, where design and implementation decisions are taken. “Play time” is the actual time playing with the instrument, where the instrument is evaluated in real-time in terms of how the playing of the instrument feels, what possibilities and sense of freedom it gives. Experience from playtime is then used again to change aspects of the instrument in design time, with multiple iterations until the performer is satisfied. Even after one or more concerts, one can take a step back to design time to adjust characteristics of the instrument.

Performing the instrument is thus an important part of the process of designing the instrument—performing is the exploration of the possibilities of the instrument

and gives inspiration on how to improve the instrument for the performance that the creator has in mind.

Waisvisz (1999) advocated to stop development at some point and take a step back from the building process and start playing with the instrument as it is: to compose, perform and explore and exploit its limitations.

In my performance “Wezen—Gewording” however, I explicitly make all the mappings and sounds changeable during the performance. At times it feels for me like switching between “design” and “play” mindsets during performance, at other times I embody the instrument and feel in control of the sound while moving, but at the same time plan what changes I want to make the next time I am at the laptop. These changes are both concerning the composition and the instrument. At the same time, having the sensors coupled to sound parameters influences how I type and move my hands over the keyboard while changing the code during performance.

It may seem that the instrument could be defined as the combination of sensors, electronics and software, and that the instrument is independent of the composition it is created for. But while this may be true for artists that search for new instruments to perform their (existing) music with, others create new instruments for particular pieces and do not necessarily use the same instrument for other compositions. In this case the instrument’s design is highly entangled with the composition itself—it is not used outside of the context of the composition it was created for. In certain cases, e.g. in Chikashi Miyama’s work “Angry Sparrow” (Miyama 2008), the mapping of the instrument even changes over the course of the composition. There, borders between composition and instrument are hard to distinguish, as the functionality of the instrument changes over the course of the performance.

4.3 Performers and Their Instrument

Where human-computer-interaction (HCI) is dedicated to interfaces that allow users to control a machine, instrument builders/performers are interested in a more intimate connection with their instrument. They aim to achieve a certain state of flow, where their instrument becomes an extension of their body. Waisvisz (1999) described this as:

During inspired performances I have experienced that a mental/physical state can emerge where a fast closed loop establishes itself between the musical intention, the muscular effort and actions, the mechanical response and the sonic feed back and the perception of this whole loop. This happens so fast that one seems to act immediately in sound and not in terms of sound and not in terms of control. Composition/performance melt into a single state of emerging, timbral, expression.

Wessel (2006) suggests the term *babbling* for the process of “non-goal-directed variation of the control parameters” as a “key to the exploration of an instruments

potential for musical expression”, it is the process of learning which sensory-motor actions will produce which kind of sonic output. Babbling seems to be an important step toward eventually feeling that the instrument is becoming an extension of the body.

The relationship between performers and their instrument is often described by themselves as **conversations or dialogues**; their instruments gain a certain amount of agency. Waisvisz (1999) is describing the relationship to his own instrument as an **intimate** one and his relationship to his sounds like that of a puppeteer to his puppets, dealing with gesture as a life-giving force. Similarly, Holzer (2011) describes the various relationships of the artists writing about their instruments as personal relationships, involving both intellect and emotion. Tudor comments (Kuivila 2004; Austin 1989):

In my electronics, I work with an instrumental principle. (...) They become my friends. They have personalities, that only I see, because of my use of them. It's an act of discovery. I try to find out what's there and **not to make it do what I want but to, you know, release what's there.**

5 Algorithms, Software and Coding

5.1 *Interactive Music Systems or Live Algorithms*

Rather than taking an instrumental approach, where a new instrument is built from scratch, various artists have pursued the making of interactive systems where a computer can listen, react and surprise an improvising musician (often playing an acoustic instrument). George Lewis' *Voyager* is a well-known example (Lewis 2007):

In my most widely performed piece, *Voyager*, originally programmed by me in 1987 and extensively updated since that time, improvisors are engaged in dialogue with a computer-driven, interactive improviser. A set of algorithms analyzes aspects of a human improvisors performance in real time, using that analysis to guide another set of algorithms that blend complex responses to the musicians playing with independent musical behavior. In *Voyager*, the improvised musical encounter is modeled as a negotiation between improvising musicians, some of whom are people, others not; the program does not need to have real-time human input to generate music.

What is interesting is that he terms the system a “piece”, suggesting that it is a composition, even though it is clearly designed for performers to improvise with it, and likely each performance will be quite different, depending on the musical vocabulary of the performers involved. The system is the musical idea, musicians enter into a dialogue with it during performance.

Young and Blackwell (2013) coined the term *Live Algorithms for Music* to describe such interactive music systems: “Live algorithms are an ideal concept: computational systems able to collaborate proactively with humans in the creation

of group-based improvised music.” In more detail they define the features as (Lewis 2007; Young 2005):

- a live algorithm can collaborate actively with human performers in real-time performance without a human operator
- a live algorithm can make apt and creative contributions to the musical dimensions of sound, time and structure
- live algorithms can contain a parametric representation of the aural environment which changes to reflect interaction between machine and environment.

They suggest that such live algorithms do not embody the artistic, compositional concepts of the author; they want to place the algorithm apart from the human designer of the algorithm. In contrast Drummond (2009) states that:

Interactive music systems are of course not found objects, but rather the creation of composers, performers, artists and the like (through a combination of software, hardware and musical design). For a system to respond musically implies a system design that meets the musical aesthetic of the systems designer(s). For a system to respond conversationally, with both predictable and unpredictable responses, likewise is a process inbuilt into the system. In all of the definitions discussed, to some degree, is the notion that interactive systems require interaction to realise the compositional structures and potentials encoded in the system. To this extent interactive systems make possible a way of composing that at the same time is both performing and improvising.

Drummond (2009) further argues that traditional distinctions between composing, instrument building, systems design and performance are blurring. In his discussion of Chadabe’s term *interactive composing* he writes:

Chadabe highlights that the musical outcome from these interactive composing instruments was a result of the shared control of both the performer and the instruments programming, the interaction between the two creating the final musical response. (...) In interactive music systems the performer can influence, affect and alter the underlying compositional structures, the instrument can take on performer-like qualities, and the evolution of the instrument itself may form the basis of a composition.

and he cites Chadabe: “The instrument is the music. The composer is the performer.”

By both Drummond (2009) and the LAM research network it is assumed that the algorithm (or system), once designed and written, is fixed. It is not changed during performance. The definition of live algorithms explicitly state that, once the performance has started, there is no human interference other than at the defined inputs of the system.

In both cases, the designed systems are the compositions. They describe the possibilities in which the musical output or the means of the dialogue, even if the outcome is surprising or unexpected for the performer (or even for its composer or builder).

5.2 *Music Software and Tools*

In the discussion above it is highlighted that compositional concepts are embedded into the code. The evaluation of the code is the means to perform the compositions. Clearly, coding is an integral part in the act of composition: it creates and defines the possibilities within which musical output can transform. **Commercial music software embodies certain kinds of compositional or musical concepts or styles.** Similarly, Tudor remarked (Kuivila 2004) on hardware that it was very hard to make certain commercial instruments do what they were not intended for. A certain musical style or way of thinking about music was embedded into that hardware. **Joel Ryan³ described musical instruments as embodiments of music theory** like, for example, a piano embodies the division in octaves in a well-tempered tuning system.

McLean (2008) takes a critical stance in the discussion of software, creativity and artistic expression:

This can lead to the bizarre situation where programmers make commercial software which practically generates music, and yet somehow the users of the software are seen as being more creative than the programmers. Here the programmers encode their musical style in the software, and the users go little beyond guiding the software to a destination pleasing to them. This can be seen in filters and plugins of music studio software as well as explicitly generative commercial applications such as Sseyo Koan Pro. The creativity of programmers is tapped into flattery of paying users.

It depends on the open-ended-ness of software in how far the artistic ideas that can be expressed with it are restricted by the concepts already embedded in the software. Audio programming environments such as Max (Cycling74) or SuperCollider (McCartney) are very open-ended and provide for many different music styles to be produced with it; yet there is a clear distinction between users of one or the other, and composers are attracted to one or the other depending on to what extent they can express their musical concepts in the language. Within SuperCollider many composers develop their own dialects or “systems within systems” and extensions to the language to “add both new possibilities and new constraints” (Rohrhuber et al. 2011). Rohrhuber et al. (2011) also discuss the “blurring of the distinction between a tool and its outcome, an application and an artwork or a model” and state that “thinking within a given language, some ideas may never occur”. From this it becomes apparent that **not only the programming language that is used to achieve a particular musical result is important, but also the programmer’s personal capacity (or vocabulary)**, the dialect she can speak within that language determines what musical ideas may be expressed.

³During his talk during the Musical Organics Symposium, STEIM, Amsterdam; May 5, 2016.

5.3 Livecoding

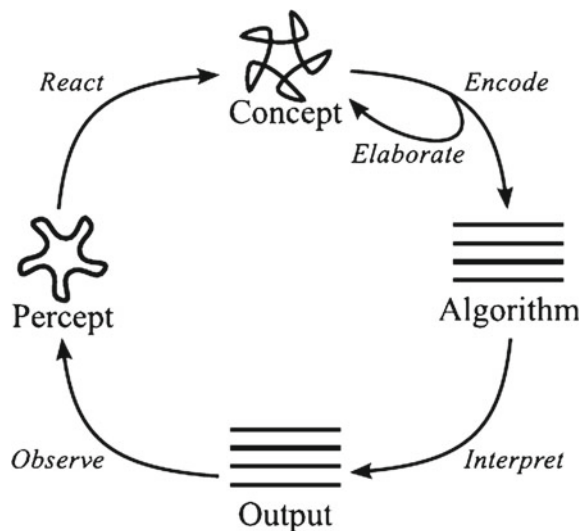
McLean (2011) describes the process of *bricolage programming* (see Fig. 5) as “a creative feedback loop encompassing the written algorithm, its interpretation, and the programmer’s perception and reaction to its output or behaviour.” In his descriptive example, he shows how the original concept of the artist is changed by evaluating the code and reacting to the output it generates. The resulting composition or code is clearly shaped in the dialogue between the artist and the code she writes and the output of the machine that runs it. McLean (2011) states: “At the beginning, the programmer may have a half-formed concept, which only reaches internal consistency through the process of being expressed as an algorithm.”

In the livecoding movement, bricolage programming is not only done to achieve a fixed code which can then be used as an instrument or live algorithm to play with, rather design time and play time is brought together into the live performance. Livecoding is a form of improvisation that would fit the definition of *instant composition* of the Amsterdam based “Carpet Collective” (Carpet Collective 2015):

INSTANT COMPOSITION combines the notion of working from the moment (INSTANTaneous creation) with the intention to build something (COMPOSING a piece with an audience present). This means that for us, improvisation principles are always concerned with both the question of FREEDOM and the question of STRUCTURE.

However, whereas with most improvisation the intended structure is not apparent to the audience, until it unfolds, with livecoding—as the audience can read the code that is usually projected in a livecoding performance—the structure can be apparent before it unfolds, and the livecoder may even decide to not let it unfold,

Fig. 5 The process of action and reaction in bricolage programming (image from McLean (2011), used with permission)



and change it again before it does. As DelaHunta (2001) wished for, **the invisible part of the performance is made visible**.

Livcoding then blurs the distinctions between composing, instrument building, and performing (and composition/instrument/performance) even further, as the computational system or code is no longer fixed once it is designed, but is adapted during the performance. A livecoder can act on all levels: changing the development of the music over time, redefining its local structure, generating events at particular times, and changing the texture of a sound while it plays.

6 Conclusion

In this article, I have reflected on my own work process, particularly in the piece “Wezen—Gewording”, asking how it affects notions of composition, instrument design and performance in the context of art works using technology such as electronic circuits and software or code.

Looking at literature from artists working in the field, the boundaries between these notions have been blurring, and there are several diverging interpretations of what is what. The views on what an instrument and what a composition is seem to overlap—if we compare the statements of Behrman and Mumma (compositions existing purely in hardware), Waisvisz (calling the mapping of digital instruments the main artistic area) and the discussion on interactive music systems.

Ryan called an instrument an embodiment of musical theory; **the instrument defines a set of possibilities that can be explored. A composition can then maybe be viewed as a particular path through this field—a particular exploration of these possibilities and music theory. The performance of this exploration makes it experiential both for the performer and listeners**—it is the event where the exploration comes to life and is translated from being a concept to being a physical, but ephemeral, event.

With the livcoding practice, the building and conceptualising of a system is transferred into the moment of performance. The distinction between design time and play time is disappearing. **As performers manipulate their instruments at the core of their functionality, they take away the notion of an instrument having a predefined behaviour**. With code that can be changed and compiled on the fly, the design of an instrument, or the making of a composition, becomes a fluid process and a performance in itself.

The interplay between the processes of composing, instrument building and performing is an embodied process along the definition of Varela et al. (1991) for embodiment. This process of embodiment is very personal, and an ongoing process. This means that **there are no fixed compositions or instruments**, and artists each have their unique artistic expression. Nonetheless they can communicate about their processes and engage with other artists to influence each other’s vocabularies.

References

- Austin, L. (1989). David Tudor and Larry Austin: A conversation. <http://www.emf.org/tudor/Articles/austin.html>. Accessed August 16, 2016.
- Baalman, M. A. J. (2016) Sense/Stage—A low cost, open source wireless sensor infrastructure for live performance and interactive, real-time environment. <https://docs.sensestage.eu>. Accessed August 16, 2016.
- Carpet Collective. (2015). Interdisciplinary instant composition: To define, share and develop concepts of improvisation. <http://www.instantcomposition.com/>. Accessed August 16, 2016.
- Cycling74. Max. <https://cycling74.com/products/max/>. Accessed August 16, 2016.
- DeLaHunta, S. (2001). Invisibility/corporeality. http://www.noemalab.org/sections/ideas/ideas_articles/delahunta.html. Accessed August 16, 2016.
- Drummond, J. (2009). Understanding interactive systems. *Organised Sound*, 140(2), 124–133. doi:10.1017/S1355771809000235
- Hayles, N. K. (2012). *How we think: Digital media and contemporary technogenesis*. University of Chicago Press. ISBN 9780226321424.
- Holzer, D. (2011). Schematic as score: Uses and abuses of the (in)deterministic possibilities of sound technology. <https://web.archive.org/web/20110506065819/>, <http://vagueterrain.net/journal19>. Accessed August 16, 2016.
- Kuivila, R. (2004). Open sources: Words, circuits and the notation-realization relation in the music of david tudor. *Leonardo Music Journal*, 14, 17–23.
- Lewis, G. E. (2007). Live algorithms and the future of music. *CT Watch Quarterly*, 30(2), 19–24.
- Maxfield, R. (1963). *An anthology of chance*. La Monte Young and Jackson Mac Low: 965 Hoe Avenue, Bronx 59, New York, USA.
- McCartney, J. Supercollider. <http://supercollider.github.io/>. Accessed August 16, 2016.
- McLean, A. (2008). Live coding for free. In A. Mansoux & M. de Valk (Eds.), *FLOSS + Art* (pp. 224–231). Goto10 and OpenMute. ISBN 978-1-906496-18-0.
- McLean, A. (2011). *Artist-programmers and programming languages for the arts*. PhD thesis, Department of Computing, Goldsmiths, University of London.
- Miranda E., & Wanderley, M. (2006). *New digital musical instruments: Control and interaction beyond the keyboard* (A-R ed.).
- Miyama, C. (2008). Angry sparrow. <http://www.chikashi.net/>. Accessed August 16, 2016.
- Nilsson, P. A. (2011). *A field of possibilities: Designing and playing digital musical instruments*. PhD thesis, Göteborgs Universitet.
- Oxford Dictionaries. (2016). Oxford dictionaries: Composition. <http://www.oxforddictionaries.com/definition/english/composition>. Accessed August 16, 2016.
- Richards, J. (2011). Lead and schemas. *Institute of Contemporary Arts: Roland Magazine*, 9. http://www.dirtyelectronics.org/docs/ICA_Roland_09.pdf. Accessed August 16, 2016.
- Rohrhuber, J., Hall, T., & De Campo, A. (2011). *Dialects, constraints, and systems within systems*. Cambridge, MA: MIT Press.
- Schnell, N., & Battier, M. (2002). Introducing composed instruments, technical and musicological implications. In *Proceedings of the 2002 Conference on New Instruments for Musical Expression (NIME-02)*, Dublin, Ireland, May 24–26, 2002 NIME02-01.
- Varela, F. J., Thompson, E., & Rosch, E. (1991) *The embodied mind: Cognitive science and human experience*. MIT Press.
- Waisvisz, M. 1999. Michel Waisvisz on gestural controllers. <http://crackle.org/MW's%20gestural%20round%20table.htm>
- Wessel, D. (2006). *An enactive approach to computer music performance* (pages 93–98). Studio Grame: Lyon, France. <http://cnmat.berkeley.edu/system/files/attachments/An+enactive+approach+to+Computer+Music+Performance.pdf>
- Wikipedia. (2016). Musical composition: Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Musical_composition. Accessed May 15, 2016.

- Young, M. (2005). What is a live algorithm. <http://homepages.gold.ac.uk/michaelyoung/LAMweb/whatis.htm>. Accessed August 16, 2016.
- Young, M., & Blackwell, T. (2013). Live algorithms for music: Can computers be improvisers? In *The oxford handbook of critical improvisation studies* (Vol. 2). Oxford University Press. doi:[10.1093/oxfordhb/9780199892921.013.002](https://doi.org/10.1093/oxfordhb/9780199892921.013.002)