Phonological and Musical Loops in Live Coding Performance Practice

DAVID KIM-BOYLE

This paper explores how various phenomena of working memory are actively drawn upon and can provide useful insights into live coding performance practice. The author argues that the phonological coding processes that cognitive psychologists believe underscore how auditory information is retained and recollected in working memory can enrich our understanding of live coding performance practice, where loop-based processes often provide key structural units. The author suggests that such practice finds structural coherence and aesthetic value through a real-time play with the effects of working memory.

In 1974, cognitive psychologists Alan Baddeley and Graham Hitch proposed a model of working memory, the fundamental principles of which have received considerable evidential support through wide-ranging experimental observation [1]. The original model consisted of two components, the visuospatial sketchpad and the phonological loop, responsible for the processing and encoding of information in the visual-spatial and speech domains respectively, each of which work under the direction of a third component, a central executive function. The model was extended by Baddeley in 2000 to include an episodic buffer that helped explain experiments in which information stored in long-term memory was retrieved for processing in working memory, including the phenomenon of chunking, whereby long sequences of information are grouped into shorter chunks [2]. The extended model is presented in Fig. 1.

Of these various components, the phonological loop is the only one directly related to the storage and retrieval of auditory information. While originally intended to model speech processing and retention, Baddeley has suggested that the basic concepts at play may underscore a listener’s comprehension of music [3] and, to that end, the applicability of the model in musical domains has received intensifying investigation [4,5].

To the extent to which the phonological loop is inherently recursive, it has the potential to offer unique analytical insights into live coding performance practice, in which generative loops and recursive and iterative functions are of fundamental structural importance.

THE PHONOLOGICAL LOOP AND ITS EFFECT ON WORKING MEMORY

Baddeley and Hitch’s phonological loop consists of two stages: a passive short-term storage that acts as a temporary storage buffer for incoming speech and an active articulatory subvocal rehearsal and recoding mechanism in which stored information is internally repeated until it is phonologically encoded. The relationship between the two is illustrated in Fig. 2.

The phonological loop helps account for a number of working memory phenomena that have received widespread experimental investigation. It has been shown, for example, that the inherent difficulty in accurately remembering words of considerable length [6], known as the word length effect, is explained by the limited temporal capacity of a phonological short-term storage facility optimized to store sounds typically not exceeding a duration of 1.5–2.5 seconds [7]. Larsen, Baddeley and Andrade have shown how phonological similarity, whereby similar-sounding words or particular sequences of phonemes are more difficult to remember than those of great contrast, is accounted for by the subvocal rehearsal mechanism of the model [8]. Interruptions to this rehearsal process also help to explain the impact of articulatory suppression and other suppression effects on the accurate retrieval of information [9,10]. Two other important phenomena explained by the model include those related to presentation modality [11], whereby the visual reinforcement of auditory information assists in the accurate retention of information, a phenomenon usually associated with Paivio’s “dual-coding” theory [12], and that of chaining [13], in which the associative relationships between sequential auditory stimuli are established during the rehearsal stage through the attribution of different weightings, a critical process for the accurate recollection of the order of phonemes within a word or numbers within a sequence, for example.
While phenomena such as articulatory suppression, word length effects, phonological similarity, chaining and modality effects are explained by a phonological loop model of working memory, it is important to remember that the phonological loop and the vast majority of experiments designed to investigate its explanatory robustness have focused on phonological encoding and various phenomena associated with speech retention. As previously noted, while there is a growing body of research investigating the usefulness of the phonological loop in helping to account for musical memory, there has been considerably less investigation into the usefulness of the model for helping to develop insights into musical forms and performance practices such as live coding, in which loops and other iterative functions are important structural determinants. Given the computationally based metaphor of cognitive models of memory more generally, it is also somewhat germane perhaps to consider the extent to which phenomena associated with a loop-based memory model can become active performance techniques.

**LIVE CODING AND THE PLAY OF MEMORY**

While live coding encompasses a wide range of performance practices, programming techniques and languages, generative algorithms, modified during a live performance, are typically of central importance. This importance is underlined by the tendency in many live performances to project the code for the audience to view. Given the important role of loops in much live coding practice, to what extent might current practitioners actively draw on some of the phenomena of working memory associated with a phonological loop cognitive model in helping to structure performances, and more interestingly perhaps, to what extent might the inhibition of memory through exploitation of these techniques be directed toward aesthetic ends?

Experiments designed to test the ability to memorize a musical phrase have shown that memorization is facilitated or inhibited by the number of notes the phrase contains and the complexity of the associative relations between those notes. Pembrook has shown, for example, that phrases longer than eleven notes can rarely be remembered accurately, while numerous experiments have shown that a random distribution of pitches, for example, are typically much more difficult to memorize than a series of pitches that have ordered, associative relationships such as those defined by tonal progressions. The length of a musical phrase that can be accurately retained by the listener is a direct manifestation of the word length effect, while associative links between pitches directly recalls principles of chaining. While musical loops can theoretically be of any length, those created by generative algorithms can be more accurately retained by an attentive listener if they contain fewer than 11 unique events. In the generative algorithm for Nick Collins’s Algoravethm 11311, programmed in the popular SuperCollider platform, for example, arrays that contain data for pitch sequences, rhythmic patterns and various other musical parameters typically do not exceed eight items (see Fig. 3).

While prescribing a low limit on the amount of data contained in an array facilitates the listener’s retention of that information, accurate retention is also supported through the phenomenon of phonological similarity. While distinctions in pitch within a looped sequence of sonic events are perhaps the most direct manifestation of phonological similarity, distinctions can be made through any number of means. In Klipp AV’s (Nick Collins and Frederik Olofsson) ICMC2007 live coding performance for example, timbral differentiation between sonic events is far more pronounced, as noncontiguous samples are recursively recalled from an audio buffer, producing fractured sonic textures with constantly shifting timbre. Conversely, looped processes may be disguised and memory inhibited by maximizing phonological
similarity as in Thor Magnusson’s *Threnoscope* [23], which will be examined later.

The Australian live coding performance duo aa-cell (Andrew Brown and Andrew Sorensen) actively explore principles of chaining in many of their performances. This usage can range from the use of Markov chains and other probabilistic distribution functions, to weight pitch selections in diatonically organized scale degree arrays, through to the use of linear functions to organize temporal structure [24]. Associative relationships need not necessarily be contained within one sonic property, such as pitch ordering for example, but may be established across auditory domains. In one example by aa-cell [25], control data used to control the timbral evolution of synthetic drones is also mapped to rhythmic patterns that gradually evolve during performance. Baddeley and Hitch have argued that the phonological loop retains associative relationships through a connectionist-type series of weightings ascribed to the sequence of events reinforced during the rehearsal process [26,27]. The manipulation of these weightings, such as occurs during a live coding performance, arguably forms the primary performance technique employed to provide structural coherence in aa-cell’s performances.

Just as the word length effect, phonological similarity and chaining may provide organizing principles for live coding performance practice.

Fig. 3. SuperCollider code excerpt from Nick Collins’s Algoravethm 11 [36]. (© Nick Collins)

Fig. 4. The graphic interface for Thor Magnusson’s *Threnoscope* [37], in which different colored bands represent the strength and duration of harmonic components of a drone. Parameters associated with these components are adjusted during performance through a script interface (right). (© Thor Magnusson)
performance practice, modality effects also help reinforce accurate retention through the multisensory presentation of information [28]. At the same time, modality effects offer a means of underscoring musical coherency. Modality effects are most commonly drawn on through the visual projection of live code and the transparency through which that code and its manipulation may be understood by an audience or observer [29]. Magnusson’s *Threnoscope*, for example, offers a live coding system in which a circular graphic interface provides a direct visual correlation to scripted code entered in a parallel window; see Fig. 4. The temporal evolution of the sound spectrum is mapped to movement around the circle, with different colored bands providing a direct correlation to harmonic complexity.

Magnusson’s *Threnoscope* is unusual among live coding systems in that it affords central structural importance to the manipulation of timbre rather than discrete pitched events. Moreover, as a drone-based system, it adopts a particular approach to temporal structure in which discrete rhythmic units are not afforded structural valence. While the loop-based nature of the system aligns it with the phonological loop’s structural model, it is the phonological similarity effect that provides the basis for its affect. Despite its cyclical temporal structure, cycles of the looped drone are made more difficult to perceive through a timbral structure that blurs discrete start and finish times of any looped material. This is particularly intensified through a timbral continuity that inhibits the perception of discrete sonic units.

To the extent to which the visual presentation of code reinforces for the listener the development of a live coding performance, any cross-modal suppression effects, in which retention is inhibited by cross-modal interruption to the subvocal rehearsal process, would seem to be reduced [30]. In most live coding performances, the cross-modal relationship of the visual to the sonic is generally intended to act as a support, a concrete memory or perceptual tracing [31] of an ever-evolving musical process rather than an inhibition.

While the visual artifact, whether in script form or graphic, explicitly provides the opportunity for an audience to better understand how a live coding performance develops, it also serves as a critical reminder for the coders or performers themselves of the musical structure they have developed and are in the process of developing. This necessarily entails that the code be clearly organized according to established conventions, efficient and intuitive in order to facilitate real-time editing [32] (see Fig. 5).

**CONCLUSION**

Phenomena such as word length effects, phonological similarity, modality effects, articulatory suppression and chaining strongly suggest a phonological loop model of working memory. Given the strong evidential support for such a model, the potential for the musical application of these effects to provide unique ways of developing structure in performance practices such as live coding, in which loops are of primary structural importance, is of particular interest. In this respect, the play between musical coherency and the constraints of memory forms an aesthetic locus for this practice. Given the cognitive model underlying many programming languages [33], live coding can be viewed as the embodiment of a *coupled process* [34], in which code becomes an *active externalism* of an internal cognitive process. Conversely, it is also worth noting the degree to which the type of code and coding processes employed predisposes live coding toward particular types of musical structure. Ultimately, the relationship between code, memory, cognitive processes and musical intention becomes a symbiotic one, where the play of and the active play with memory create unique musical forms with rich potential for further development.

---

**Fig. 5.** Script excerpt from Sorensen’s *Strange Places* [38], in Sorensen’s *Impromptu* [39] programming language. (© Andrew Sorensen)
References and Notes


6. Note that accurate retrieval of words with many phonemes is affected more by a word’s temporal duration than by its phonemic complexity.


11. Salame and Baddeley [10].


13. Baddeley and Hitch [7].

14. Thompson and Yankelov [4]; Berz [5].


17. Popular languages include ChucK, EMacs, SonicPi and SuperCollider, but other languages such as ixi and Impromptu have been developed by live coding practitioners themselves.


26. Baddeley and Hitch [7].


32. Brown and Sorenson [24].

33. Ormerod [14].


35. Baddeley and Hitch [1].


37. Magnusson [23].


Manuscript received 2 January 2017.

DAVID KIM-BOYLE is an Australian composer and new media artist. His research and creative practice is primarily concerned with the representation of complex, nonlinear forms with real-time graphic notations. Further information may be found at <www.davidkimboyle.net>.