Live Algorithms

Tim Blackwell

A Live Algorithm takes part in improvised, collaborative performance, sharing the same modes of communication and expression as its partners. Autonomous rather than automated or controlled, the device enjoys the same constraints and freedoms as its human associates.

Ideally the machine improviser conforms to the established practice of a medium, capable of imitating and developing shared gestures, behaviours and ideas. Furthermore, the live algorithm would be expected, at times, to contribute novelty and surprise, to experiment and take risks, and to assume leadership. Other performers experience the live algorithm as if it were a human, with a sense of validity and belief.

This idealised concept of a live algorithm was articulated by the Live Algorithms for Music (LAM) research network, building in the work of a small number of researchers who have pioneered autonomous music systems. LAM is concerned with a particular type of live algorithm, those that are capable of performing free musical improvisations with other musician. (It is understood that the improvisations may be vocal but do not involve words.) This musical domain is ostensibly an easier route for the study of performance algorithms since bodily presence, movement, facial expressions etc. have only secondary importance, and the prohibition of language, as is common in this domain, apparently liberates the system from vast semantic problems.

The aim, it is stressed, is not to reproduce the capacity of a human performer. Instead, we wish to explore a mechanical, algorithmic world, expanding our own potential for music making.

Although designing a live algorithm with the ability to imitate and develop shared ideas is already a formidable undertaking, the additional requirement of innovation is an even harder research challenge. Without the capacity to innovate, however, the audience and the performers would lose the belief that the live algorithm was truly engaged with the performance and not merely accompanying it. We suggest that it is the ability to innovate that distinguishes autonomy from automation and randomness.

Dagstuhl Seminar Proceedings 09291
Computational Creativity : An Interdisciplinary Approach
http://drops.dagstuhl.de/opus/volltexte/2009/2221

Although we wish to access the iterative and the generative, outputs that are too mechanical and predictable, or alternatively too random and impenetrable would work against our aims. The reconciliation between the unpredictable and the mechanistic is hard to achieve. Although it is an open question as to whether innovation can derive from rule following, we sidestep the issue by focusing on *interactive* performance. We postulate that in such a performance, novelty and surprise can be explained as an emergent phenomenon. The system, which comprises several, maybe many, individual performers, self-organises micro-level events into macro-level structures. The structures are not scripted and are not explicit at the level of individual participants who themselves may not even know that the structures are forming

The minimum requirements for emergence are unknown. In the lack of a workable understanding of self-organisation, one route forward is by the modelling of systems where we know emergence occurs. These are chiefly the self-organising systems prevalent in the natural world, especially those with emergent animal behaviour, since these latter systems are the richest. The aim is to use natural analogies to set up artificial performance systems which contain the potential for emergence.

To this end, current live algorithm research focusses on certain *open dynamic systems* which model some aspects of a natural system in which emergence is known to occur. These live algorithms contain a parametric representation p of the appropriate external environment (an environment populated with sonic and musical events, for example). System state x is driven forward by low level rules $f(x, p)$. The system is computationally open since it is not known in advance which one of the vast number of time dependent input series $\{p(t_1), p(t_2), p(t_3) \ldots\}$ will occur. State is itself interpreted as parameterised actions $q = Q(x)$ on the environment. The modules which effect these transformations, P, Q, and F, or at least their logical equivalents, are arguably necessary sub-functions of any live algorithm. Loosely speaking, P and Q correspond to analysis and synthesis functionality, with F embodying inner, hidden processes that are only accessible by interpretation.

Within this model, interaction is defined as a contingent change of system state; it is anticipated that the state space must be large enough so that new states previously unrealised may be induced. This strengthens weaker models of interaction in which human-computer interaction is essentially selection from a menu.

The model is based somewhat on the stigmergetic interaction of social insects. Stigmergy, as originally observed and noted in the study of insect organisation, is the process by which individuals interact indirectly (i.e. not via contact or immediate messaging) by making changes to their immediate environment. Spatial proximity between interacting insects is therefore not essential for communication. The environment retains an imprint of previous encounters and is accessible to any passing individual. The modifications can be regarded as messages responsible for mediating the social processes of the swarm.

The entire range of performative possibilities constitute the relevant environment of a performer, and therefore of a live algorithm. At its lowest level a performance can be envisaged as a sequence of physical events. This sequence is meaningless unless organised. Order, however, is imposed both by the output of a single performer (modelled by the dynamic rule f) and by the self organisation of outputs of several performers (modelled by stigmergy).

This environment is mapped by the live algorithm designer to an open state space. The actions of other performers and of the live algorithm are represented as states in this space. The key difference is that the motion of the image of external events is determined by interaction, and not by an algorithm. State flow of the internal state is however algorithmic; it is informed, but not dependent on p.

Some differences between people and dynamical systems are immediately evident. Memory enables performers to revisit past actions and understand relationships; evaluation, followed by learning, leads to improvement; a social context provides encouragement and criticism and a cultural context imparts meaning via a web of shared experience.

However, a biological system may exhibit some of these features. The 'simple' ant is considerably more sophisticated than its point-like analogue in a dynamical system, possessing many more internal degrees of freedom, and insect society is considerably more complex than the sketch presented here may imply. Closer modelling may enhance the ability of our artificial systems; for example pheromone trails provide a mechanism for reinforcement learning.

We speculate if a live algorithm culture could be created, and if this is the missing ingredient. Perhaps improvisers could communicate successful (and unsuccessful) experiences, gossip, compete and reward each other. Perhaps the machine culture should be allowed to develop separately and in isolation from human culture for a while. Or perhaps machine and human culture could mix: can we imagine sending a live algorithm to music college? And how important is the social context; should be deprive our live algorithms of the opportunity to experience pride when performing well, and, if performing badly, shame?