Methodological Framework and Design Process for Applying Evolutionary Simulation to Musical Interactions

Insook Choi

Studio for International Media and Technology, MediaCityUK, University of Salford, Greater Manchester, UK

i.choi@salford.ac.uk

Abstract

This paper focuses on a methodological framework where the creative design process evolves through iterative cycles. The design process undertakes a complex network of tasks for integrating two domain models: dynamical simulation and musical interaction. The framework accounts for engineering technical and compositional affordances to accommodate evolving behaviors to be expressed in real time performance interplay. This is illustrated with a case study of simulated swarms of heterogeneous agents. Highly integrated parallel work streams are elucidated with sub-process elicitation in simulation, system integration and software engineering, composition, and performance. Framework formalization draws upon the established RAD model with significant modification to present the extended version that can be multi-threaded for concurrent creative processes. Two landmarks of 20th century music automation are drawn diachronically to frame the technical discussion in a social context of listening practice, developed by modeling creative process and testing musical assumptions. Revisited cannon is redirected from bygone exemplars to ongoing practice, illuminating three baseline requirements for a methodological framework: interdisciplinary platform architecture, complex systems model of music creation, and agile listening. Concluding theses on second order listening and interdisciplinary architecture summarize the proposed methodological framework addressing contextual listening and technical culture.

Introduction

A computational simulation of multi-agent evolutionary dynamics is potentially highly compatible for application in music. While the mid-20th century afforded innovation in music with a post-WW2 industrial revolution, the 21st century has yet to see a very different landscape in music practice. The methodological framework presented here aims to promote compositional practice with multi agent evolutionary dynamics as one of the prevailing investigations of 21st century creative endeavors. While the diverse research objectives can be articulated within this framework, the focus will be on a creation framework with an emphasis on design process methodology. For implementation details on subsystems, readers are directed to (Choi and Bargar 2013; 2014) where algorithms, sound engines, parameter mapping, and statistical analysis on emerging patterns are extensively presented.

The compatibility between an evolutionary simulation and music can be best described in terms of temporal dynamics. Evolving temporal dynamics is inherent in the systems that simulate multi agent interaction. Working with temporal dynamics from evolutionary simulation brings forward the implication to conceive music performance as an evolving interaction where the interaction is *generatively situational*, meaning the performance evolves through the immanent dispositions that are generatively produced by autonomous agents in constant flux; the performer joins and influences the flux bringing a new dimension to the generative music requiring highly responsive and agile performance in situ.

In this generatively situational state, the system keeps a human performer engaged in constant observation presenting challenges to the performer to check her expectations and assumptions of the system behavior and consequences of interaction, which adds a gameplay-like yet highly reflective quality to the performance. The simulation incorporated for this work (Sayama 2007; 2009; 2010) is not yet tailored for crafting its outputs to known and educated aesthetic expectations. For this researcher, who is an experimental composer, this is more a blessing than a problem. But perhaps, this is where we should start.

Method conveys a consistency of practice; framework conveys the terms of that consistency. In this paper, in order to distinguish method from methodological framework, references are drawn diachronically from historical context to preface technical discussion with a fresh take on listening practice applied with automation. Juxtaposition of historical and technical topics illuminates the framework discourse to help readers see the detailed technical discussions

The Fifth International Workshop on Musical Metacreation, MUME 2017.

www.musicalmetacreation.org.

not as isolated method but rather as a case study of listening perspective brought to technology. Here reframing of historical perspectives may go against cannon, and in so doing may validate the shift to create an alternative space for juxtaposing perspectives and technical studies.

Critical Reflection on Historical Framework

The prospective visions and dreams of music computation in the 20th century generated abundant experiments and new methodologies, and surely brought about diversity in configurations of, both the tools to compose with and the ensemble to perform with. However, a large part of the contemporary music scene is still deeply rooted in responses to two 20th Century innovations, serialism and *musique concrète*. The tradition referred to as the Darmstadt School advanced the democratization of membership of audible attributes in sets of discrete units, such as twelve-tone scale. Extended organization of sets of discrete auditory units led to computer applications for music data management, yet acoustic foundations were still anchored on the tuning of the 18th century's well-tempered clavier.

Pierre Schaeffer's musique concrète simply bypasses the tuning tradition inherent to musical instruments by working with recordings of found sound sources. For most novice listeners, while serial music might have sounded like harsh dissonances one after another with no kind entrance to appreciation, musique concrète might have offered some way in due to the simplicity of the sound events and sequences, in addition to the listeners' ready familiarity of art forms such as photography, radio, and cinema. In contemporary practice musique concrète techniques have been reconceived as a palette of signal analysis, filtering, convolution and resynthesis. For a remarkable example see the comprehensive summit of computational music concrete reflected in Risset's *Sud* (Risset 1985).

Computational processes made for much more efficient procedural sound synthesis, sampling, algorithmic composition, and automatic score generation. Whole compositions can be built by hierarchical organization of subroutines with fine control of sound using processes such as stochastic distribution, fractal or cellular automata, or other nonlinear dynamics that could yield ranges of tones perceivable with meaningful features, and also yield time organization of an entire piece with holistic treatment.

Then what is missing? Perhaps nothing. The better formulation of an inquiry can be on illuminating what remains to be challenging and underexplored with computational potential while we might have been holding on to perspectives conditioned in some kind of time capsules. Achievements in modeling musical style from Cope to Dubnov for more than a quarter century (Cope 1991; Dubnov and Surges 2013) offered new insights on learning from a corpus or constructing a database, of which research results impacted on generative techniques for music. Cypher (Rowe 1993), introduced with clean and transparent articulations by Rowe, pioneered machine listening applied to music with the justifiable benefit of MIDI events despite inherent signal loss, bringing a constructive approach to building the layers of critiques over musical parameters. Kinetic Engine (Eigenfeldt, 2006) is a multi-agent environment where agents' personality traits evolve ensemble rhythmic interaction. Social descriptors are assigned to musical behaviors while agents' judgments on similarity converge on polyphony and heterophony.

What remains curious in the history of computer music is the obscured identity of Lejaren Hiller. While it has been widely taught that he was the first composer of algorithmic music using a computer output in traditional music notation (Holms 2012) such acknowledgement is far less significant than his actual contribution towards future practice, which is little known. To partially elicit his contribution, the section below briefs a comparative case study of the first opuses of Hiller and Schaeffer.

Schaeffer and Hiller: Novelty Revisited

In 1948, repurposing WW2 electronic equipment Pierre Schaeffer created *Étude aux chemins de fer* (Schaeffer 2010). With a series of subsequent etudes, what Schaeffer demonstrated was musical expression and what he composed was musical structure with real-world acoustic events, therefore 'concrete' music. Schaeffer's experiments were focused on the mechanized reproduction and manipulation to achieve musical properties. The terms, "concrete" and "abstract," are adapted to create a dialectic distance between two worlds of sounds, the world of non-music and the world of music. A lesson from this creative activist was clear: one can find potentials from any concrete sounds, even noises; then excavate musical elements through processes; then compose a structural presentation to demonstrate musical expression. Automation that enables systematic and reproducible transformations of sound is necessary to carry out this program.

In 1956, utilizing the Illiac (Illinois Automatic Computer), a first-generation mainframe computer, Lejaren Hiller created *Illiac Suite for String Quartet,* which was premiered in August of the same year (Hiller and Isaacson 1957). Lacking idiomatic expressions, *Illiac Suite* is often poorly received other than by those who are unusually curious. What Hiller demonstrated was an experiment and what he composed was a process of imposing a set of rules and order to unordered initial states. This experimental approach bears full fruit in the contemporary example, Tipei's *Many Worlds* (Tipei 1998) for percussion ensemble, which achieves a fine balance of computational outcomes in instrumental realization.

Schaeffer and Hiller create two ends of a spectrum of seminal approaches in the history of automation applied to music. What are the lessons to be learned? As an experimental composer, I give some care to refresh my performance as a listener in pursuit of evolving perspective on the relationship between computation and creativity to guard against my own habit as a musical protégé. The following formulate the determinants and the factors that shape the innovation and the culture of listening:

- Critical encounter/Interdisciplinary exposure: Schaeffer and Hiller had access to machines and facilities, and careers in non-musical domains, Schaeffer in broadcasting and engineering, Hiller in chemistry and molecular modeling. They pursued musical aspirations.
- Modeling a Creative Process: The difference between Schaeffer and Hiller's examples is clear and superficial compared to the common practice they share, conducting early experimentation in modeling creative processes. Schaeffer modeled musical properties from everyday sounds; Hiller modeled the acquisition of musical properties.
- Testing musical assumptions: These experimental works resulted in two listening cases. By modeling musical properties from everyday sounds, Schaeffer generated a case where listeners recognized both new and familiar. By being consistent with algorithmic processes, Hiller generated a case where listeners recognized the string quartet but did not recognize "music".

Étude aux chemins and *Illiac Suite* merit revisiting through time due to their simplicity and transparency. Focus comes with the unexplored interdisciplinary setting juxtaposing the model of machine and apparatus with the model of music; the experiments had to be simple and focused, which allow the act of composition to be heard. Regarding creative process, the different approaches in the two opuses draws an important distinction between modeling musical properties by transformation of audible source material and the acquisition of musical properties by statistical methods. Regarding musical assumptions, the two opuses set side-by-side are significant as they yield dialectic insights to aesthetic expectations.

Baseline Requirements for a Framework

Translating the above points as reframed lessons for contemporary relevance, we can further elaborate and develop them to arrive at three baseline requirements for a methodological framework for applying complex systems to music. 1) **Interdisciplinary platform architecture** – with methodologies for integrating a scientific simulation into a music creation environment. 2) **A complex systems model of music creation** – Schaeffer extended the methodology of music creation by new techniques such as groove, variation on speed and play directionality. The methodology was new and compatible with the mechanical operations enabled by the recording and playback system in his time. Hiller extended the methodology of music creation by new techniques such as stochastic algorithm for introducing music informatics process with permutable unit model of musical elements. The methodology was new and compatible with computation enabled by Illiac, the first mainframe generation computer in his time. Situating evolutionary simulations in music creation requires a methodology to conceive the music creation framework as a complex systems model that integrates concurrent processes among *simulation, sound production engines, interaction* and *instrumentation*. A set of distinguishing design directives has to be articulated because all music creation systems are traditionally complex. The directives guide the integration strategies (Choi and Bargar, 2014), as the model requires provision of requisite variety in sound making to the variety in evolutionary dynamics. 3) **Agile listening** – a performance framework for kinesthetic interaction with a simulation while paying close attention to self-organizing trajectories through and by listening. The most important factor for the performer engaged in this framework is a performance of listening through which she has to find her performance trajectory that would evolve along the evolutionary trajectory of the simulation she is interacting with.

Simulation and Methodology

A platform we refer to as *Wayfaring Swarms*, was developed to integrate Sayama's heterogeneous multi agent simulation (Sayama 2007; 2010) into a performable configuration. Agents are simple 2D entities defined as a center point and radius, moving in a bounded plane. A number of agents (usually 100 to 300) are initialized at random positions and velocities. Motions are dynamically influenced by social engagement, assigning each agent a perceptual radius and rules for interactions. Within an agent's perceptual range the control parameters are: Cohesion: an agent moves toward the average position of local agents; Alignment: an agent moves towards the average velocity of local agents; Separation: an agent avoids collision with local agents; Whim: an agent moves randomly with a probability; Pace Keeping: each agent approximates its speed to its own normal speed. Agents stray randomly when they do not perceive others.

A heterogeneous swarm has multiple subgroups of agents defined as unique types, each with a unique set of control values. All agents perceive one another and respond non-uniformly according to type, resulting in rich emergent behaviors. New agent types may be spawned by asexual perturbation and mutation, or by mixing parents' types, a method similar to crossover (Unemi 2003).

First Iteration of Platform with Analog Capacitive Sensing

Wayfaring Swarms was first implemented using an analog surface (48-inch x 36-inch) with capacitive sensing. The table (x, y) coordinates are calibrated with the agents' navigation space in the simulation, and the visualized simulation was projected down onto the surface while the performer's hand movement was sensed through capacitive surface to generate a time series of (x, y) data values. Hand movement data then was passed to the simulation where it was perceived as another agent's behavior.

The resolution of performance data to simulation is determined by the spatial and temporal resolution of the system response to a performer's hand movements. The reliable resolution of position sensing for touch interaction was an area of roughly 0.3 inches square with fuzzy boundaries, approximating a differentiable resolution of 160 x 120 units on the 48 x 36-inch table. A 0.3-inch unit constrains resolution capacity between performance signal and system response time resulting in signal losses from hand movement, which requires performer's awareness to either let go or compensate over time. The positions of simulated agents provided a higher resolution since it was projected on the table surface bypassing capacitive sensing. A computer graphic image of 1024 x 768p scaled to 48 x 36 inches produces a unit of about 0.05 inches/pixel. The ratio of this resolution to the rate of change of agents' movements determines the rate of change of sound control data.

To summarize this iteration, the analog capacitive surface for a signal path resulted in too much signal loss due to inadequate spatial resolution to pass the hand movement data controlling the simulation, which has much finer spatial resolution of the agents' data controlling sound. Further, the 10Hz to 15Hz data rate of touch interaction was slower than the internal simulation time step of 60 Hz that updates the agents' states and transmits control data to sound. The spatial and temporal resolutions described above influenced the composition of *Mutandrum* (Choi 2010) by reframing the limitations and losses into compositional strategies for, 1) the selection of sound palette, 2) the level of musical structure progressing as tableau, and 3) the phrase level of gestural organization where transformations were applied instead of fine control over individual acoustic events. Further technical discussion can be found in (Choi and Bargar 2013; 2014).

Second Iteration of Platform

The second iteration of the Wayfaring Swarms platform was implemented using a touch screen with 16:9 ratio and resolution of 1920 x 1080 pixels. With a screen area of 21 x 12.5 inches, it is 0.01 inches per pixel, a greater resolution by a factor of 5 over the first iteration. The high resolution digital touch screen, reduced in size from the analog by 2x in width and 3x in height yet provided a greater access to navigation area of simulation agents' movements. Time step in 60 Hz was applied uniformly to screen touch responsiveness, simulation state update, and control data transmission to sound. To summarize this iteration, the smaller performance surface with higher spatial and temporal resolution enabled more efficient gestural control for managing a number of agents and for influencing agents' distributed behaviors, resulting in more engaged interaction with agents. The improvement on resolution offered more in-depth opportunities for the composition, *Human Voice* (unpublished; see https://vimeo.com/206638013): 1) in increased choices of sound synthesis and mapping, 2) in more progressive and developmental musical structure, and 3) in more flexible and fine gestural control both in macro- and micro-duration.

Simulation and Integration to Architecture

Working with a complex system, structuring software design and developmental cycles is critical to support creative processes and to be guided by them. Version updates must not be too disruptive for the processes occurring in parallel such as learning and testing the simulation, then its rapid turnaround to compositional planning and performance trials. Any changes during version updates must be clearly communicated and immediately tested to maintain highly integrated circularity of all co-evolving processes. The progression of the Wayfaring Swarms platforms was conducted using a Rapid Application Development (RAD) process (Naz and Khan 2015). Using agile methods and working from core performance architecture, function modules were tested and refined in an iterative cycle. This process was dynamic and reflected on the requirements generated by compositional criteria. The modules were developed to provide 1) data translations and mappings, 2) control streams management between devices, 3) feature extraction, and 4) synchronized scheduling of event initializations and transitions. Compositional criteria and musical ideas were used to shape an initial set of general requirements then performance trials were used to elicit further requirements and specification as well as to demonstrate the musical ideas to drive platform enhancement. Musical ideas were tested often for feasibility, resulting in refinement, or revision, or sometimes setting aside.

This approach is alternative to a waterfall method where a platform is completed before composition begins. Rather than composing a musical work linearly from beginning to end, the work was composed through iterative prototyping of an initial end-to-end design. An extended RAD model was developed (see Figure 1) from the original diagram in (Martin 1991). The prototypes both in software and composition were reused and subjected to further refinement towards a final product. Function-based iteration enabled a manageable cycle of decoupling, upgrading, and reconnecting function modules, as requirements were refined, so that composition continued with available modules while others went offline. Module requirements were assessed according to three work streams: performance design, technology design and composition. Work streams were integrated across a phased composition methodology, described in the following section.

Composition Methodology

The compositional tasks are articulated in three phases. These phases define requirements across the work streams, performance design, technology design, and composition. The work streams constitute parallel design processes to elicit, test and apply system requirements. Each phase links compositional subtasks to one or more work streams. The order of subtasks in each phase indicates the priority of the associated requirements. For example, in Phase 1 the first subtask generates performance design requirements, the second subtask generates technology design requirements, and the third subtask generates composition requirements. This order of subtasks indicates the performance design requirements drive the technology design, which in turn feeds to context aware composition requirements. In Phase II, a composition-oriented subtask takes priority. The requirements cycle over three work streams indicates how criteria of one work stream impact the others.

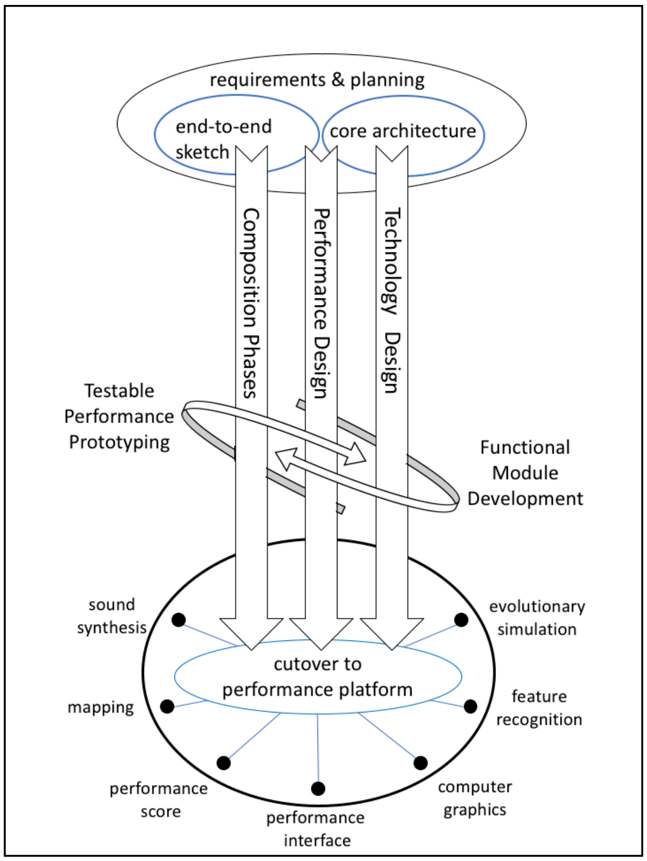

Figure 1: Generalizable RAD for evolutionary music creation. The sub-systems in cutover plane vary from project to project.

Phase I: Pre-Composition Phase with the simulation

This phase involves three subtasks:

1) Resource exploration of evolutionary model: Investigate the makeup and variety of simulation constituents and explore constituent types by varying perceptual and behavioral parameters. Agents' types are programmable as a set of behavioral and perceptual parameter values for groups of agents in the format of $X * (R^i, V^i_n, V^i_m, c^i_1, c^i_2, c^i_3, c^i_4, c^i_5)$, where X number of agents is declared first followed by the attribute values for perceptual range, velocity norm, velocity max, cohesive force, separating force, random steering probability, and pace tendency. The output of this subtask provides performance design requirements: Evolutionary behaviors determine the performability of the simulation.

2) Combinatorial exploration: different combinations of types are explored to discover agents' interaction behaviors and emergent pattern formations. Heterogeneous multi agents' interaction can be designed by a recipe with N sets of parameter values, which activates groups of N different types of agents into one initial state. The output of this subtask provides technology design requirements: Ensure that data of emergent behaviors can be generated and captured during performance, and ensure that a performer can designate and introduce changes to simulation parameters during a performance.

3) Mapping experimentation: while exploring types and combinatorial behaviors, possible choices of sound synthesis methods are imagined and prototyped. The curation of agent types and their combinations is narrowed during mapping experimentation. At this stage, a maximum variety of initial states of a given simulation set up must be tested with chosen mapping scenarios because the sound response can be very different with different initial states. The output of this subtask provides composition requirements: The composer either curates population sizes of agent groups and combinations of agent types, or adjusts sound synthesis parameter values then tests again.

Phase II: Composition Phase with the simulation

This phase involves three subtasks:

1) In-depth learning and curation: learned from exploration of simulation, the mature understandings of possible emerging patterns and a range of self organizational behaviors. Understanding temporal dynamics is critical to project a musical form at large from local dynamics. The output of this subtask provides composition requirements: Identify a repertoire of classes of potential evolutionary events.

2) Refining the design of agents' types and final curation: prior to engaging with various performance scenarios, it is advisable to choose final candidates of types. The output of this subtask provides composition requirements: Possible musical form is planned to make sure the wide range of variety is available in the candidates.

3) Performance Scenario: this is perhaps the most important part of the composition. Articulation of performance scenarios comes at this stage before sketching out the musical form of the piece. Working with an evolutionary simulation, compositional task can be better described as an architectural and design task, which aims at an optimal blueprint for performance, designing constraints, and solving problems of performance interface layout. Designing constraints also aims at reducing performers' cognitive overload in addition to acting on musical elements. Performance here is not exclusively about executing a composition, rather about engaging with the simulation along

with the compositional plan, which is expressed in designed notation as a blueprint for performance.

This subtask provides performance design requirements: Performance scenarios are organized as an end-to-end blueprint for the musical piece. Performance is defined as a rendering mode of the blueprint where each instance of renderings yields variety within designed constraints. At the core of a performance scenario is an act of listening to guide the play. In addition, a gamut of manual repertoire can be designed for performer interaction with agents' group behaviors such as stretch, merge, separate, lead, and scatter clusters of agents.

Phase III: Composition Phase with Iterative Sketching of Musical Form

Suggestive musical form undergoes revisions through phase II. While scoping out variety of performance scenarios, a form is imagined and hypothesized. To prepare for rehearsals phase III finalizes the form in terms of return structure of elements. Elements that can be consulted are:

- Agent types and their combinations – assessed by temporal dynamics and interaction behaviors. Selecting the numbers of types to present yields insights to the overall duration of the piece. From the total pool of types to work with, the similarity or contrast in behaviors yields insights to musical form.
- Sound synthesis – where and when applied in a musical form. The choice of sound synthesis is to be made based upon the discoveries from phase II. The resulting perception of synthesis method is an important criterion, as it constitutes auditory experience to guide both performer and listeners.
- Behavioral mapping between simulation and sound synthesis – engineered for sonifying agents' behaviors. When designing agents' types in simulation, it is recommended to plan a list of sound synthesis engines to test with. At the risk of being reductive, following are two examples. When cohesive force among agents is strong, frequency modulation technique is a good choice since the varying parameter value in simulation behavior is slower and subtle, which reflects well in dynamic spectral profile with modulating frequency covariation. When cohesive force among agents is weak and agents' velocities and movement variations fluctuate within wide range, granular synthesis technique might be a good choice as the ranges of jitters in simulation can be reflected in the ranges of jitters in collective grains resulted in density shifts across multiple frequency bands.
- Layout of Performance Interface – designed as navigation space. Perhaps this concept is among the most unique in that it can be described as a new proposition anchored on technical consideration. When working with simulation through direct manipulation, the screen space becomes a landscape that a performer must manage during live execution (see Figure 2). The frequency of how often the performer engages and disengages functional controls and how these are accessed while attending to agents must be figured in the musical form.
- Conveyance of simulation features - to elicit feature data to control audible features. Evolutionary features may be unmeasurable by simulation data and require feature analysis modules to be added.

Phase III outputs: the first three elements above generate composition requirements; the Performance Interface element generates performance design requirements; the Feature Conveyance element generates technology design requirements.

The iterative cycle of Phase III typically unfolds as follows: 1) Estimating the duration and number of sections, then subtracting or adding more types by redesigning the properties of agent groups. 2) After examining similarity and contrast of behaviors, subsections are sketched out. 3) Further sketching with behavioral mapping and sound generator refinement determines which section requires how many number of subsections

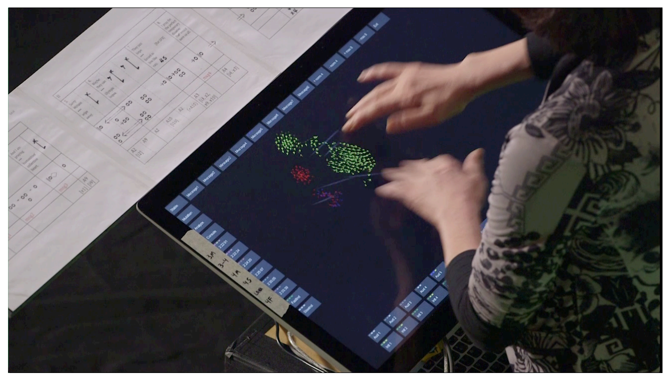

Figure 2: Evolutionary music performance surface for engagement with agents, and a portion of a performance score.

Performance with Heuristic Observations

Evolving features and patterns are signatures of complex systems. They are the result of emerging collective self-organizational behaviors. While each agent and each type of agent follows a set of simple behavioral rules, the mix of different behavioral rules yields nonlinear and unexpected formation of clusters or dispersions. A performer's main task is to generate a live, coherent musical structure working with simulated agents, which exhibit repelling or attracting movement with varying speed based on the state of the collective force.

These temporal dynamics require continuous performance interaction at micro-duration level (beats; seconds) as well as long-duration intervals (phrases; minutes). Sometimes an evolutionary system exhibits native coherence the performer cannot override and must find the path to join with the composition intent. At the core of heuristic observation is listening. A blueprint or performance score can be provided to function as a map of references but it is critical that the performer learns to anticipate behavior by listening to the state of the simulation. The estimated duration of a piece defines when to progress to the next state of the simulation or next section of the piece, and when to prolong a current state or section. The moment of "do nothing" is discovered in rehearsal and in performance. Working with an evolutionary model a heuristic decision consults both the performance score and the emerging features in simulation.

For system methodology, soft requirements for heuristics in performance are incorporated and articulated here:

- Intervention: to facilitate manipulation of an evolutionary system a performer relies on auditory features to convey transient qualities in the simulation. For example, while steering a cluster of agents to merge or stretch, the rate of agents' functional separation or joining is telegraphed in sound. The transient quality of auditory features helps govern a performer's pace of hand movement.
- Prolongation: to suspend the tendency of a simulation to move towards a basin of attraction and settle into a state of equilibrium. Sometimes a performer may prolong a state of equilibrium or the state of transients while acting or counteracting an attracting or repelling force.
- Do nothing: to let the simulation do the work. There are enough inherent dynamics in a complex system, which unfold into a rich variety of evolving patterns with audible signatures to give a performer clues when to choose to wait and do nothing.
- Local Trials: Sometimes an initial condition does not provide an adequate variety a performer aims to achieve in a certain section – so an affordance is provided for the performer to re-initialize the state without interrupting the local audio events or the ongoing musical form. This recurrence is perceived as a musical repetition with variation. Having this in mind the performer need not hesitate to iterate and re-initialize by heuristic decision.

Performance Interaction Methodology: manual and cognitive tasks

To support soft requirements the methodological framework develops space in a composition to facilitate performance with heuristic observation. This space is represented in the performance score and reflected in the design of the performance interface. Performers' actions are needed to generate example listening conditions to drive the RAD cycle to refine technical modules for the Wayfaring Swarms platform requirements. Module functionality is represented in the graphical layout of the performance interface, and in the visualization of objects where performers' actions are imparted. Functional attributes of the performance surface are developed through the composition phases, and require optimal management of screen real estate to avoid space overlap and conflict of control icons in regions for interacting with evolving visualization.

Intense cognitive tasks are on demand to evaluate when and how to interact with the simulation, and when to touch control icons and in what combination. As in Figure 2 the center of interactive performance space is largely used to display visualized simulation. Three edges of the space are lined with classes of function control icons: to initiate different initial conditions, to change message passing to synthesis parameters, and to call different sets of agents to sound synthesis mapping structures. The pixel resolution and aspect ratio of the performance surface are embedded as initial conditions in the spatial dimensions of the simulation. This enables performance surface attributes to impact the evolutionary behavior. Dexterous manipulation carries the iconic function – in the sense of Peirce's semiotic triad icon-index-symbol (Peirce 1903), which is directly analog to perceiving the impact of the manipulation on the behaviors of the simulated agents.

Lessons learned from and inadequacy of direct manipulation for complex tasks has been well critiqued in research literature (Flohrich 1997; Buxton 1993). A degree of indirection is designed in the signal path from simulation data to synthesis engine to construct an orchestration so that not every visual event must correspond to an audio cue. Here dexterous manipulation carries an indexical function, where agents' behaviors generate complex sound events and transformations not feasible via direct manipulation. The iconic function supports an intuitive relationship for manipulating simulated agents, while the indexical function supports a learned relationship for guiding agents through an audible evolution governed by heuristic observation. Sounds' and agents' configurations change and develop through a composition and the platform and musical structure provides space for a performer to acquire musical associations by learning evolutionary behaviors and unlearning fixed associations of agents and sounds.

Conclusion

The selected opuses from Schaeffer and Hiller stand out in history, even from their larger bodies of work, indicating traces of novel perspective in automated musical creativity, through discovery of transformation processes applied to musical modeling and statistical acquisition of musical properties. Heuristic observation of data from evolutionary computing can identify similar traces that will connect to paths of musical creativity, providing alternatives to modeling history of musical styles or performers' behaviors.

The conclusion of this paper can be summarized with two theses. One addresses listening culture, the other addresses technical culture. *1) Second order listening is like an agent performing a listener, whose role is to elicit the processes or the act of the work of art beyond its acoustic phenomena or known aesthetic judgment.* Perhaps, established aesthetical criteria or learned judgment may not be sufficient for understanding creativity. This applies to both human agents and machine agents. At the core of computational creativity is machine learning. Unless accompanied by unlearning, especially with novel experimentation described in the previous section, our understanding of creativity will be extremely limited therefore leading to non-creativity. *2) An interdisciplinary architecture for applying evolutionary computational simulation to music must facilitate both the horizontal integration across disciplines and the vertical integration over workflows from composition, technology, and performance design.* When working with evolutionary systems, a true paradigm shift cannot be achieved by simply plugging simulations into a music toolbox. These two theses can be partially validated by conceptualizing a 'listening score', for example. In the Wayfaring Swarms platform the concept of listening score emerged through the following: 1) the evolutionary dynamics are visualized, 2) the visualization becomes a dynamical score for performers to interact with, and then 3) it is displayed for an audience as a listening score.

Technical requirements for this methodological framework can be summarized: 1) a parameterized evolutionary simulation implemented to generate temporal dynamics observable in real-time; 2) a configurable performance interface for simulation control and data visualization; 3) an efficient authoring tool for data mapping and routing from interface to simulation and from simulation to sound synthesis; 4) an end-to-end performance platform connecting 1-3 above for rapid prototyping; and 4) an extended RAD iteration method multi-threaded with versioning tools for co-evolution of performance design, technology design and composition.

Heuristic observation is a method required in RAD and in performance, and is rehearsed. A performance score can represent this requirement by illustrating the balance of direct and indirect manipulation in the sequence and relative duration of a composition's states and events. A performer's indirect control of detailed sound generation through social interaction with simulation agents provides a temporal framework for heuristic observation. Performance of co-evolving social engagement is where and when the role of second order listening, performing a listener is enacted.

References

Buxton, W. 1993. HCI and the inadequacies of Direct Manipulation systems. *SIGCHI Bulletin*, 25(1), 21-22.

Choi, I. 2010. *Mutandrum*. Sensors and Information Processing for Musical Expression, Concert (April 2010), Voorhees Theater, Brooklyn, New York. https://vimeo.com/74618886

Choi, I. and Bargar, R. 2013. Between Music and Games: Interactive Sonic Engagement with Emergent Behaviors. In Reidsma, D., Haruhiro, K., and Nijholt, A., eds. *Advances in Computer Entertainment, Lecture Notes in Computer Science*, Vol. 8253. Heidelberg: Springer, 519-523.

Choi, I. and Bargar, R. 2014. Sounds Shadowing Agents Generating Audible Features from Emergent Behaviors. In Sayama, H. et al., eds. *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*. Cambridge: MIT Press, 726-733.

Cope, D. 1991. *Computers and Musical Style*. Madison, WI: A-R Editions.

S. Dubnov and G. Surges, 2013. Delegating Creativity: Use of Musical Algorithms in Machine Listening and Composition. In Lee, N. ed., *Digital da Vinci*. New York: Springer, 127-157.

Eigenfeldt, A. 2006. Kinetic Engine: Toward an Intelligent Improvising Instrument. *Sound and Music Computing Conference (SMC),* Marseille, 97-100

Flohrich, D. 1997. Direct Manipulation and Other Lessons. In Helander, M., Landauer, T. and Prabhu, P., eds. *Handbook of Human-Computer Interaction*, Chapter 8. North Holland: Elsevier Science Publishers B.V.

Hiller, L. and Isaacson, L. 1957. *Illiac Suite*. New York: Theodore Presser Co.

Holms, T. 2012. *Electronic and Experimental Music: Technology, Music, and Culture*. New York: Routledge.

Naz, R. and Khan, M.N.A. 2015. Rapid Applications Development Techniques: A Critical View. *International Journal of Software Engineering and Its Application* 9(11)s: 163-176

Martin, J. 1991. *Rapid Application Development*. Macmillan Publishing Co., Inc., Indianapolis, IN, USA

Peirce, C. S. 1903; 1931-35. Hartshorne, C. Weiss, P. and Burks, A. eds., *Collected Papers of Charles Saunders Peirce*, Vol 2. Cambridge, MA: Harvard University Press, 243-52.

Risset, J-C. 1985. *Sud • Dialogues • Inharmonique • Mutations*. Compact Disc Recording. Marseille: INA-GRM – INA C 1003.

Rowe, R. 1993. *Interactive Music Systems*. Cambridge: MIT Press

Sayama, H. 2007. Decentralized control and interactive design methods for large-scale heterogeneous self-organizing swarms. *Advances in Artificial Life.* New York: Springer, 675-684.

Sayama, H. 2009. Swarm Chemistry. Artificial Life, 15(1): 105-114

Sayama, H. 2010. Robust morphogenesis of robotic swarms. *Computational Intelligence Magazine*, IEEE, 5(3):43–49.

Schaeffer, P. 2010. Étude aux chemins de fer. In *5 Études de bruits - Études aux objets*. Vinyl Disc recording. Disques Dreyfus – FDM 46050 36286 1. Paris: Disques Dreyfus – SB 775.

Tipei, S. 1998. Many Worlds. In *Raw Cuts*. Compact Disc recording. Urbana: Computer Music Project, University of Illinois Experimental Music Studios.

Unemi, T. (2003). Simulated breeding–a framework of breeding artifacts on the computer. Kybernetes, 32(1/2):203–220.