Musical Metacreation: Papers from the 2012 AIIDE Workshop
AAAI Technical Report WS-12-16

Quantum Composition and Improvisation

Dale E. Parson
Kutztown University of Pennsylvania, P.O. Box 730, Kutztown, PA 19530
parson@kutztown.edu

Abstract

Quantum mechanical systems exist as superpositions of complementary states that collapse to classical, concrete states upon becoming entangled with the measurement apparatus of observer-participants. A musical composition and its performance constitute a quantum system. Historically, conventional musical notation has presented the appearance of a composition as a deterministic, concrete entity, with interpretation approached as an extrinsic act. This historical perspective inhabits a subspace of the available quantum space. A quantum musical system unifies the composition, instruments, situated performance and perception as a superposition of musical events that collapses to concrete musical events via the interactions and perceptions of performers and audience. A composer captures superposed musical events via implicit or explicit conditional event probabilities, and human and/or machine performers create music by collapsing interrelated probabilities to zeros and ones via observer-participancy.

Introduction to Quantum Musical Systems

One walks into the concert hall with a sense of potentiality. Anything could happen. Typically, some number of parameters for the upcoming performance have been bound in advance, and even advertised, thereby shaping expectations. Still, music is partly about novelty in a unique performance environment. The prelude to the prelude is anticipation.

This position paper is an outline of a perspective of musical system as quantum system. Adoption and exploration of this perspective can contribute to the richness of composition, performance, listening, and to their interaction. This perspective suggests constructive approaches to the representation of a composition as a superposition of probable events, of instrument design as providing means for exploring the superposition, and of performance as a collapse of the superposition into perceivable musical events. The availability of analysis-synthesis informational feedback loops across human and machine performers is intrinsic to this perspective.

A quantum system is a highly generative system that generates behavior when an observer-participant exchanges information with the system in the form of question-answer pairs, wherein question and answer are two sides of the coin of interaction (Wheeler 1988). Posing a question collapses a subset of the superposed states, cascading to collapse related superposed states, in the limit cascading to the point where the system emits a concrete answer. Posing an answer extends the superposed states of the system. Questions are answers in that they expose the quantum state of the questioner, thereby entangling that state with the system, thus extending the system. Answers are questions in that the cascading collapse of superposed states uncovers additional superposed states. From a musical perspective, asking a question consists of exploring the superposed state of the composition, and retrieving an answer consists of collapsing a subset of the explored superposed states, thereby generating musical events. A musical system wherein a performer can sense the composition and the composition can sense the performer – "sensing" being the interactive exchange of the question-answer coin – is a generative system that is extensible at performance time. Improvisational extensibility is a primary goal of this work.

Definition of some terms is in order. In a quantum system a *superposed state* is a simultaneous overlay of alternative, complementary classical states, where a *classical state* is a state that can be measured by an observer, for example a musical event that can be perceived by a listener. Only one classical state can be perceived to exist by an *observer*, who forces collapse of a superposition into a classical state through an act of *measurement*. "Measurement tends to disturb the system measured." (Lloyd 2006). Various seeming paradoxes in the folklore of quantum mechanics arise because in some cases the classical states are conflicting states that the quantum system takes on simultaneously. States can be *entangled* – conditionally interrelated – and an observer becomes entangled with a system by way of measurement. Some quantum physicists go further in characterizing measurement as *observer-participancy* (Wheeler 1988). Measurement is never a strictly passive activity, since it disturbs the system by collapsing some of its superposed states, at the same time making measurement of other state values inaccessible. Observer-participancy extends this historical perspective by viewing the observer as part of the quantum system. In an interactive system an observer-participant can extend the quantum system by contributing superposed states to the entanglement. This interactive extensibility is the core of Wheeler's 1988 proposal.

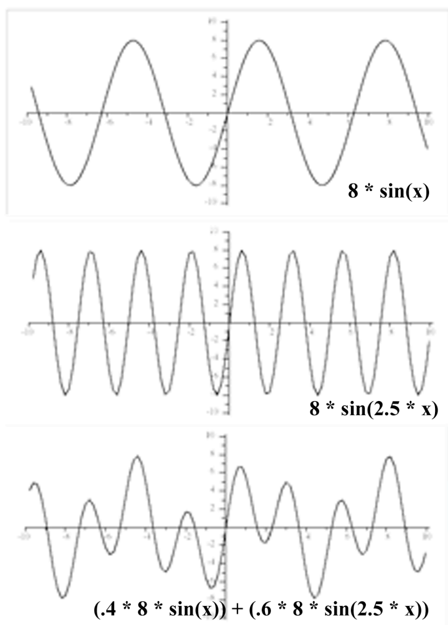

Figure 1: A linear superposition of waves

Figure 1 illustrates a simple linear superposition of two sine waves. The respective weights of .4 and .6 in the superposition represent probabilities of measuring the respective waveforms. The contributing waves are mutually exclusive in this example, so that an observer measures only the 8 * sin(x) wave at a probability of 40%, or the 8 * sin(2.5 * x) wave at a probability of 60%, when interacting with the system. The superposition exists only in the unmeasured state. Quantum mechanics uses complex-valued wave functions and so-called *probability amplitudes* that do not concern us here. The essential points are that the quantum system holds a superposition of states, with analysis characterizing each contributing state with a probability function, and that the measurement of the state by an observer collapses probability of one of the available states to 100% and the others to 0%.

We are dealing in discrete events and their discrete probabilities. Suppose at a certain point in performing a composition, there is a 100% probability of the system emitting a C note, a 100% probability of emitting a G note, and a superposed probability of emitting (40% E versus 60% Eb). It is evident to the musical reader that, at a higher level of structure, there is a 40% probability of emitting a C major triad and a 60% probability of emitting a C minor triad. Furthermore, while it is true that the lower-level performer action of selecting Eb cascades to the collapse of the chord to C minor, it is equally true that the performer action of selecting a C minor chord cascades to the collapse of the note being considered to Eb. States are superposed at hierarchical levels, and collapse to a 100% or 0% concrete state at one level of structure can cascade to dependent states across multiple levels of structure. Composition captures a sequence of superposed event states, and performance takes the form of collapsing some superposed states to 100% or 0% of their component states, thereby cascading and emitting music, while leaving other superposed states intact. Conventional musical notation takes this approach to its limit by setting probabilities within the score near or at 100% or 0%.

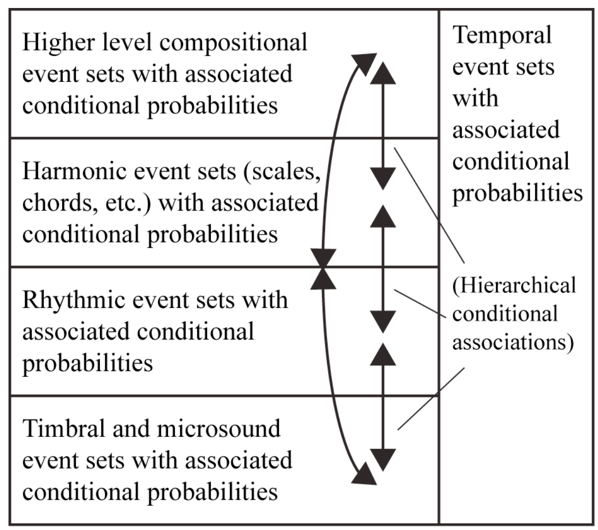

Figure 2: Hierarchical events with conditional probabilities

This discussion suggests a bidirectional, hierarchical network of interrelated musical events and their associated conditional probabilities as a means for storing a quantum musical score in a computer. Figure 2 illustrates the idea. Sets of mutually exclusive, superposed musical events at one level of compositional hierarchy may, upon resolution to 100% or 0% probability, cascade collapse of associated event probabilities at other levels as indicated by the arrows. Analysis and synthesis of microsound, timbral, rhythmic, harmonic, and higher order compositional structures occur at respective temporal granularity as suggested by the vertical set of temporal event sets and their conditional probabilities on the right.

While such a representation of musical score is possible, the use of a probability-oriented network need not be explicit. The section below on concrete field studies shows that it is possible to use stochastic representation schemes that do not employ explicit conditional probabilities.

We end this introduction by considering Kauffman's comment on extensibility as generative power in a Boolean network (Kauffman 1995).

> Our intuitions about the requirements for order have, I contend, been wrong for millennia. We do not need careful construction; we do not require crafting. We require only that extremely complex webs of interacting elements are sparsely coupled. (p. 84)

Kauffman's statement and accompanying examination of intra-network coupling and its relationship to static stability, cyclic stability, non-repetitive generative behavior and chaotic behavior – the four classes of activity of so-called *complex generative systems* – dovetails with Wheeler's use of the generative growth of the telecommunications industry and its participants as an example of a self-synthesizing system. It dovetails equally well with the proposal to use an explicit or implicit network of interrelated musical events and their associated conditional probabilities as the basis for musical score and improvisational performance. Under-coupling in Kauffman's networks leads to rigid, simplistic system behavior, and over-coupling leads to chaotic behavior. Part of the art of designing a quantum musical system is the art of adjusting the coupling between superposed state hierarchies so that their collapse into musical events come under the creative control of composers and performers.

Related Work

Stochastic composition and statistical musical analysis have a substantial history. Roads gives an outline and a set of references for the former (Roads 1996). Cage (Cage 1961) and Xenakis (Xenakis 1992) are noted for their groundbreaking work in this area. Loy outlines a set of probability-based compositional strategies (Loy 2006). Temperley explains various statistical techniques for musical analysis including probabilistic models, Markov chains and Bayesian models (Temperley 2007).

The primary contribution of the present work to this legacy is architectural. Descriptions and models of quantum systems use probabilities and probability amplitudes, but quantum systems are not composed of collections of probabilities. A quantum system does not carry a set of probabilities in its pocket. A proposed quantum musical system is a system for the capture and improvisational performance of a musical score using superposed musical events that performers collapse into concrete musical events at performance time, collapsing subsets of mutually exclusive event sets into sounds or silence. In a real sense all performing musicians do this already, and the current proposal is an effort to formalize that architecture and to embody it in a set of tools. The use of conditional probabilities is a technique for implementing such an architecture. The core thrust here is to provide a model and means by which to capture a superposition of hierarchical musical events as a computer-based score, and to allow performers to create cascading collapses of superposed events into concrete events, while at the same time extending the superposed score at performance time. Performance time extensibility via performer actions and system adaptation is key. The next section outlines three experimental systems that have contributed to this position.

Three Concrete Field Studies

Analysis of Real-time Finger Picking

The initial inspiration for the present position grew out of a debugging session for chord traces in a system that analyzes and extracts musical structures from a live, finger picked MIDI guitar's data stream (Parson 2006). That system makes no use of probabilities, but system behavior is nevertheless nondeterministic because of normal variations in articulation and timing. Non-essential details in the data stream vary in each performance.

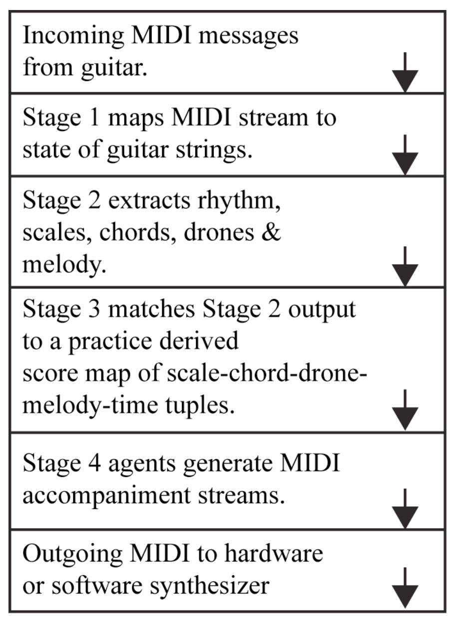

Figure 3: Pipelined system for analyzing MIDI guitar playing

Figure 3 illustrates the pipelined architecture of that system. Analysis stage 1 reconstructs the state of guitar strings with respect to pitch, amplitude, and articulation techniques such as plucks, hammers, pulls, slides and chokes. Analysis stage 2 determines meter and tempo by analyzing finger picking patterns, then it determines scale, chord and tonal center by matching collections of pitches within temporal windows established by rhythm to predetermined patterns. Analysis stage 3 matches transitions in the output parameters of stage 2 to stored, statistically analyzed traces of stage 2 output from previous practice sessions. Stage 3 composition capture in this system consists of repeated, consistent playing (so-called "practice"), followed by the averaging of traces for stage 2 output parameters. Capture is thus a stochastic process that integrates over variations in practice. Stages 4 MIDI message synthesis consists of hand-tailored code for a composition that reads the output of the preceding stages and generates accompaniment in the form of MIDI messages for downstream sound synthesizers.

The inspiration for the present position came with the realization that stage-2 chord detection never settles on the chord-as-played-in-the-intent-of-the-performer, but instead cycles through a cluster of interrelated chords that center about the chord of the performer's intent. Part of the reason is that chords in finger picking are arpeggiated chords, with notes added and subtracted in rapid succession. The faster the harmonic transitions, the more cloud-like the clustering. Even without intentional arpeggiation, though, real chords as played exhibit this effect. Fingers do not come down onto keyboard keys and flat picks do not strum guitar strings simultaneously. MIDI is a serial protocol, and so MIDI messages for distinct notes in a chord cannot arrive at a synthesizer simultaneously. Intentional arpeggiation emphasizes the effect, but the point is that all chord soundings include transitional effects that are essentially arpeggiation at the quantum level of sounding.

Figure 4 shows a sample stage 1-and-2 chord trace at the top and an abstracted chord cluster at the bottom. Numbers at the top are sequence numbers for incoming MIDI messages, followed by the MIDI note being sounded on the six guitar strings, with the bass string on the left. To the right in parentheses is the chord extracted by stage 2 for that string state. The algorithm uses a 12-place bit map for the notes being played, e.g., bit 0 for C, bit 1 for C#, and so on. It uses table lookup in a 4096-entry table to find the closest matching chord for the sounding notes in the equally tempered scale. The matched chord is the closest to the played notes, where proximity is the *Hamming distance* between the played and complete chord bit patterns, this distance being the number of bit changes needed to go from one to the other. Missing notes filled in by the algorithm in this example are tagged with "?" in the cluster when they are missing in some played examples, and are struck through when they are missing entirely from the played notes. The algorithm's selection of fill-in notes is arbitrary in the version being debugged here. It is straightforward to make both fill-in and removal of mismatching notes probabilistic, where the predominant key or some other property of the piece conditions the probabilities of filling or removing a note to achieve a chord match.

46: --- --- --- F5 --- D5	(detected as D minor)
49: --- --- Bb4 F5 --- ---	(detected as Bb minor)
51: --- --- --- F5 --- ---	(detected as ???)
53: --- --- --- F5 A5 ---	(detected as F major)
54: --- --- --- F5 A5 D5	(detected as D minor)
55: --- --- Bb4 F5 A5 D5	(detected as Bb major 7^{th})
57: --- --- Bb4 F5 --- D5	(detected as Bb major)
62: --- --- A4 F5 --- D5	(detected as D minor)

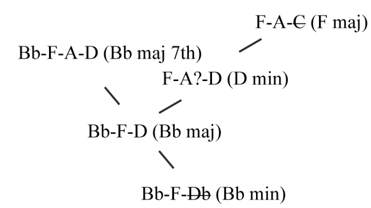

Figure 4: Arpeggiated chord as a probabilistic chord cluster

Such probabilistic adjustment of the piece's analysis and synthesis at performance time is what this proposal is getting at. Rather than capturing a fixed score with a fixed note pattern on a staff, a composer can capture a cluster of notes with associated probabilities, either in detail or from a library of composer-specific clusters. In this MIDI guitar system score capture consists of stochastic stage-3 integration over a series of practice sessions that saves consistent stage 2 output transitions and discards inconsistent ones. Human playing at performance time drives the resolution of tempo, meter, and analyzed notes in the system as they apply to scales and chords, all of which drives stage 4 accompaniment. Conversely, apparent detours through the probability space by a performing computer can lead the human performer into those detours, effecting improvisation. This system uses the human's playing to drive analysis decisions, and the player is free to follow variations in the system's response. Performance has the feel of interaction. Playing with this system exhibits cyclic reinforcement of human and computer improvisation.

Implicitly Stochastic Game-based Synthesis

The second field study is a game-to-music improvisational system based on mapping the rules and state of an on-line Scrabble™ game to synthesized MIDI music (Parson

2010). In contrast to the previous field study that focuses on analysis, this one focuses on synthesis of compositional structure. The game-to-music metaphor is that of opening up corridors in a maze as players place Scrabble words on the board. Upon placement, a virtual software composer traverses the maze and collects lists of words. The composer maps letters to notes in a scale being played on a MIDI channel – each of up to 16 MIDI channels has its own letter-to-scale mapping and other mapping parameters – and passes the translated notes to a MIDI event scheduler. This mapper determines how many letter-notes to play per beat, tempo, meter, accents, sustain, octave extent, and other generative properties, most of which are set by a human conductor via a graphical user interface.

Statistical distributions of letters and words provide a basis for mapping structures from word lists to notes, chords, and phrases. While pseudo-random tile selection provides a stochastic aspect to the instrument, players use knowledge of vocabulary to impose structure on this sequence of pseudo-random selections. The imposition by players of domain-specific structure on a set of pseudo-randomly selected game elements and the subsequent mapping of this structure to musical structure are aspects that set it apart from most previous work in chance-based music. Players impart structure, but the resulting musical structure is only semi-intentional on their part. Mapping makes the structure musical. The musical potential of a game configuration is a superposition that collapses to concrete music under the direction of a software mapper controlled by a human conductor during performance.

This game has enjoyed numerous ensemble performances and less formal interactive demonstrations at recitals, festivals, conferences and recruiting fairs. At one conference demonstration an attendee who was a self-described “expert Scrabble player” placed a block of tightly interlaced crosswords in one move, resulting in the generation of a tightly clustered, arpeggiated polychord. The musical effect, caused by repeated mapping of the sets of shared cross-letters in the tightly woven crosswords, was novel at the time, demonstrating that game skill could map to performance skill more effectively than anticipated. The game is fun to play because of familiarity with the game coupled with the novelty of fairly sophisticated music generation. The game informs the current study because the state of the game is a superposition of potential musical states with non-explicit probabilities that collapses game state to music through the actions of the players and the mapping conductor.

Explicitly Stochastic Game-based Synthesis with Live Coding

The most recent field study extends the approach of Scrabble-to-MIDI by creating a new game that allows players to manipulate explicit probabilities attached to dynamic game components via live coding (Parson and Reed 2012). The game in question, named *HexAtom*, allows players to inject large numbers of atoms of 12 element types, one for each interval in the equally tempered scale, onto a planetarium dome. The game metaphor is the expanding universe, which starts out being one atom wide; its expansion is driven by atomic motion. Figure 5 is a photograph of HexAtom at its premiere.

Figure 5: HexAtom play in an early universe

The atoms tile the dome as nested and adjacent hexagons. Individual atoms have direction and velocity, and their constituent element types have stochastic properties including reflectivity, tendency towards atomic fission, atomic fusion, ability to expand the universe at its circular boundary, and tendencies towards creating and following curvature of space (simulated gravity). Each element type property has its own probability of occurrence. Many of the probabilistic properties relate to multiple atoms (e.g., deflection, fission, fusion, and gravity), so the probabilities are conditional, based on the state, proximity and probabilities of neighboring atoms. Players can place atoms with initial trajectories as well as modify atomic properties using either a graphical user interface or live coding in the interpreted Python language. Increasing the probability of fission, for example, splits atoms into their constituent atoms with smaller atomic numbers, tending to push music generation towards the tonic – element 0 is usually the tonic, element 1 the 5^{th}, and so on in a typical scale – while increasing the probability of fusion has the inverse effect. Python live coding allows players to play “at a higher level.” Play often consists of injecting streams of atoms to create pleasing ensemble geometries on the dome. Spatial locality of the atoms being sounded by the music mapper maps to locality in the planetarium’s Surround Sound system, so that the effect is musical as well as visual.

The atom state-to-MIDI mapper is an extension of the one used in Scrabble. The conductor can manipulate it using either a graphical interface or live Python coding. Where the mapping parameters in Scrabble are static scalar and vector values adjusted by the human conductor, HexAtom allows the conductor to write Python functions that update these parameters periodically. For the conductor, Python live coding provides the ability to induce stable periodic, aperiodic, and chaotic behavior into the mapper via custom functions. As a result of live coding, players can increase the number of superposed states, increasing algorithmic information content, by moving probabilities away from 0% and 100%, thereby increasing nondeterministic atomic interactions, or decrease the number of superposed states by moving probabilities to 0% or 100%, making atom interaction deterministic. One way to end a performance is to set probability of fission for all elements to 100%, decaying all atoms over time to element 0 (the tonic) and then into dark non-existence (silence). The composer can add superposed states to the game-to-MIDI mapper by replacing static scalar and vector parameters with stochastic functions, or manually collapse them to classical states by using static values.

Conclusions and Future Work

The framework of a composition as superposition of states that cascade to music via interaction with performers has been fruitful. In the systems discussed, entanglement between the system and performers is bidirectional. The guitar analysis system includes stochastic elements for tracking a player's performance, matching it to a practice-derived score, and generating accompaniment, which the performer hears and can couple into on-the-fly decisions about improvisation. Superposed musical states latent in the letter and word choices of a Scrabble game resolve to musical state via conductor-directed mapping, and anticipated musical effects come to influence players' choices. HexAtom extends the gaming approach by introducing explicit probabilities under player control, and by providing live coding for both play and mapping, allowing performers to extend the degree of state superposition. The quantum architecture may be explicit or implicit in the compositional and performance tools. Interactive audio-visual gaming guided by the quantum perspective is especially productive.

The author became aware of related work in the emerging field of *quantum interaction* only at the time of finalizing the accepted version of this paper (Busemeyer 2012, Kitto 2008). Investigations have applied the formalisms of quantum superposition, entanglement and complex-valued probability amplitudes to problems in cognition, information retrieval, economics and emergent processes. Orientation is towards application of formal quantum techniques, including quantum probability, to highly context-dependent systems, where measurement entangles with the system and where behavior of a global system can depend on the specific sequence of interactions among its parts in violation of constraints of classic probability. At least one effort is investigating applying quantum techniques to music (Chiara, et. al., 2008). That work concentrates on applying the formalisms of quantum interaction to analysis of existing scores and performances. The present work uses quantum interaction as an informal architectural framework for constructing virtual musical instruments and scores. Future work will apply the formalisms of quantum probability and quantum modeling as elucidated in the emerging quantum interaction literature to the domains of instrument design and score capture.

References

Busemeyer, J. Quantum Cognition and Decision Notes, http://mypage.iu.edu/~jbusemey/quantum/Quantum%20Cognition%20Notes.htm , Indiana University, link tested August 2012.

Cage, J. 1961. *Silence: Lectures and Writings*, 50^{th} Anniversary Edition. Middletown, CT: Wesleyan University Press.

Chiara, Giuntini and Negri, From Quantum Theory to Music, *Advanced Science Letters* 1 (2008), 169-178.

Kauffman, S. 1995. *At Home in the Universe, The Search for Laws of Self-Organization and Complexity*. New York, NY: Oxford University Press.

Kitto, K. 2008. Why Quantum Theory? *Proc. of the Second Quantum Interaction Symposium*, University of Oxford, UK.

Lloyd, S. 2006. *Programming the Universe, A Quantum Computer Scientist Takes on the Cosmos.* New York, NY: Random House.

Loy, G. 2006. *Musimathics, The Mathematical Foundations of Music*, Volume 1. Cambridge, MA: MIT Press.

Parson, D. 2006. Real-time detection of finger picking musical structures. *Proc. of the 9th Int. Conference on Digital Audio Effects* (DAFx-06), Montreal, Canada.

Parson, D. 2010. Algorithmic musical improvisation from 2D board games. *Proc. of the Int. Computer Music Conference* (ICMC 2010), New York and Stony Brook, NY.

Parson, D. and Reed, P. 2012. The planetarium as a musical instrument. *Proc. of the 12^{th} Int. Conference on New Interfaces for Musical Expression* (NIME 2012), Ann Arbor, MI.

Roads, C. 1996. *The Computer Music Tutorial*. Cambridge, MA: MIT Press.

Temperley, D. 2007. *Music and Probability*. Cambridge, MA: MIT Press.

Wheeler, J. A. 1988. World as system self-synthesized by quantum networking. *IBM Journal of Research and Development* 32(1):4-15.

Xenakis, I. 1992. *Formalized Music, Thought and Mathematics in Music*. Hillsdale, NY: Pendragon Press.