Evan X.Merz
University of California at Santa Cruz
evanxmerz@yahoo.com

Creativity in Algorithmic Music

1. Introduction

In this essay I am going to review the topic of creativity in algorithmic music [1], focusing on three perspectives on creativity offered by three groups of composers. The first section will review the definition of creativity offered by computational psychologist Margaret Boden. The second section will examine one possible measure of creativity. The next section will look at three different composers, their attitudes toward creativity and the way their algorithms embody those attitudes. Finally, I will critically examine the core questions that are being asked by algorithmic composers.

2. What is Creativity?

Many authors have speculated on the nature of creativity [2]. It is not my goal here to offer a universal definition of creativity, or to review the numerous suggestions that have been offered previously. In this paper I will start with Margaret Boden's thorough conceptualization of creativity, and show how her definition applies to the work of various algorithmic composers.

Boden describes a few different dimensions of creativity. First she defines the context for creativity.

> "What you might do - and what I think you should do in this situation is make a distinction between 'psychological' creativity and 'historical' creativity (P-creativity and H-creativity for short). P-creativity involved coming up with a surprising, valuable idea that's new *to the person who comes up with it*. It doesn't matter how many people have had that idea before. But if a new idea is H-creative, that means that (so far as we know) no one else has had it before: it has arisen for the first time in human history." [3]

Then she describes three different ways new ideas can arise. The idea of conceptual spaces is important to Boden's definition of creativity. Conceptual spaces are thought-patterns that define the things that can be thought. As Boden defines them, conceptual spaces typically arise from a given culture or education, as opposed to arising uniquely in each person. In other words, a conceptual space is defined by how a problem is parameterized, and how that may create or abolish conceptual barriers. When she describes three different creative mechanisms, she outlines them in relation to conceptual spaces.

The first type of creativity she describes is combinational creativity. This is the simplest form of creativity and "involves making unfamiliar connections of familiar ideas." [4] She cites the examples of making a collage, creating poetic imagery or creating analogies. Combinational creativity is simply connecting two ideas from familiar conceptual spaces. For instance, when Romeo asks if he should compare Juliet to a summer's day, he's pondering combinatorial creativity. He's not exploring new territory in the english language, or changing the way english is written, he is simply putting together two concepts that might not otherwise have been connected: Juliet and a summer's day.

The second type of creative mechanism defined by Boden is exploratory creativity. As you might guess, exploratory creativity involves exploring a conceptual space. Boden argues that exploring the edges or just undefined areas of a conceptual space are a type of creativity.

> "if the style of thought is an interesting one ... then even just exploring it will lead to many novelties, and may reasonably be regarded as 'creative'." [5]

An example of exploratory creativity in language is using parts of words to create new words which are possible within the english schema, if not necessarily listed in the dictionary. In the age of the internet, this is occurring constantly. For example, exploratory creativity might have occurred whenever someone first spoke of a handheld, or a chatroom, or of the celebrity couples Beniffer and Brangelina.

The final type of creativity defined by Boden is called transformational creativity. In this version of creativity, a conceptual space is fundamentally altered by making a new connection.

> "The deepest cases of creativity involve someone's thinking something which, with respect to the conceptual spaces in their minds, they *couldn't* have thought before. The supposedly impossible idea can come about only if the creator changes the pre-existing style in some way. It must be tweaked, or even radically transformed, so that thoughts are now possible which previously (within the untransformed space) were literally inconceivable." [6]

Examples of tranformational creativity in language are more difficult to conjure up. One example might be when words from other languages bring foreign concepts to english speakers. For instance, there is no native english equivalent for the concept of *ennui*. When that word was first incorporated into english speech, it subtly transformed the space of the english language, enabling new thoughts, new exploration, and new works of art.

Boden's framework for creativity requires one other crucial ingredient: the new idea must be *good*. In Boden's conception of creativity, and idea "must be useful, illuminating, or challenging in some way." [7] Evaluation is a crucial step in this version of creativity. A new idea can't be a random smattering of elements; it must be a well-ordered, useful and interesting idea for it to be called *creative*. But evaluating creativity, in all the forms it might take, is a non-trivial task.

3. How Can We Measure Creativity?

The concept of creativity is a human concept. We think we know that humans can be creative based on our judgement that our own work, or other people's work, is creative. But what do we measure when we measure creativity? Do we judge the output of a process to determine how creative it is? In other words, is it the artifact created that matters? Or is it the underlying process that is important?

Boden argues for the latter, that the underlying process must be creative in order for the system to be judged creative. In essence, this is a cognitive view; the creative agent must simulate a process that is in some way similar to the human process of creativity.

> "whether a program models exploratory creativity depends more on its inner workings than on the novelty-value of its outputs. The crucial question is whether the output was generated by processes that explore, test, map and/or transform the conceptual space inhabited by the program concerned." [8]

Other academics argue the opposite, that the output of a system should be the sole determinant of creativity. Graeme Ritchie, for instance, argues that, when judging creativity in non-human agents, we apply an artificial standard.

> "In human creative activities, there are certain aspects which are knowable, such as the attributes of the artefact created, the other comparable artefacts in existence, possibly the other artefacts the creating individual was aware of. What we usually do

not know is the mental or emotional processes by which the individual produced the artefact (although we may know other aspects of the action, such as the time taken). Hence, it is routine to make judgements of creativity (in humans) on the basis of what is known, often focussing on the attributes of the artefact(s). If our formal definition of achieving creativity, for analyses of computer systems, is to mimic our judgements of humans, then it too should be based only on comparably observable factors." [9]

In the same paper, Ritchie then offers 14 measurements upon which the creativity of a computer program might be empirically rated. His methodology involves polling volunteers to asses the aspects of novelty, quality and typicality within a set of computer-generated output.

> "**Novelty.** To what extent is the produced item dissimilar to existing examples of its genre? **Quality.** To what extent is the produced item a high quality example of its genre? ... **Typicality.** To what extent is the produced item an example of the artefact class in question?" [10]

The subsequent fourteen questions are designed to give multiple scales on which to measure these quantities. Ritchie is clear to point out that the fourteen factors are only one way of looking at this data.

> "The aim of the original presentation of the criteria ... was to show how to make precise *some* factors which are of interest when assessing a potentially creative program, in order to illustrate a range of possibilities which would-be assessors of programs could select from, add to, or modify in a systematic way." [11]

Ritchie's framework is interesting and valuable because it provides testable criteria upon which an empirical judgement of creativity might be applied to a computer program. The major drawback to his system is that it requires test subjects who can rate a corpus of work. Hence, any application of the system requires a significant outlay of time and money; it's not a system that can easily be applied by a lone graduate student. It also creates a serious problem when considering time-based media. If you want to rate the creativity of a sonata-generating program, it would take a significant amount of time for raters to assess the body of work.

There may be drawbacks to Ritchie's framework for measuring creativity. However, the task he is trying to tackle is almost Herculean in scope. In the face of a seemingly insurmountable task, Ritchie's system allows us to ponder what we should look for in creative algorithms. It gives us a base from which we can examine the output of creative algorithms.

4. Algorithmic Creativity

In this section, I'm going to examine the algorithms employed by three different groups of composers, and relate them to Boden's mechanisms for creativity.

4.1. Charles Ames and Early Innovators

In comparison to today's technology, the pioneers of algorithmic composition were computationally limited. They didn't necessarily have the power to simulate anything that might resemble the process of human composition. Does that imply that those systems were not creative? Not necessarily. Making creative systems wasn't necessarily a concern of the algorithmic composition pioneers. In general, they were more concerned with making systems which generated music that fulfilled their own aesthetic priorities and satisfied whatever conceptual criteria were relevant for a particular piece.

The pioneers saw the computer more as a tool; an extension of their own creative process. Iannis Xenakis, for example, thought he could program his own mental maps into software.

> "A musical work can be analyzed as a multitude of mental machines. A melodic theme in a symphony is a mold, a mental machine, in the same way as its structure is. These mental machines are sometimes very restrictive and deterministic, and sometimes very vague and indecisive. ... Certain mechanizable aspects of artistic creation may be simulated by certain physical mechanisms or machines which exist or may be created." [12]

Xenakis used computers primarily to calculate values based on probabilistic models, a method he called *stochastics*. In his book he often says that a piece was *calculated* by a computer, rather than it being created or composed by a computer. For most investigators of creativity, this is a significant differentiation. After all, if a machine is just using predetermined rules to specify its output, is it really creating anything new? According to Boden's three categories of creativity, such output could only be combinatorial or exploratory creativity, but Boden cautions that the appearance of creativity is not the same thing as creativity.

> "creativity is a matter of using one's computational resources to explore, and sometimes to break out of, familiar conceptual spaces. On closer inspection, it is clear that the program did not break out of its initial search-space. It did not even bend the rules, never mind break them." [13]

This runs contrary to the empirical measurements for creativity laid out by Graeme Ritchie, which don't evaluate the process by which a piece was created.

Charles Ames has approached algorithmic composition similarly in a large number of algorithmic composition programs. Ames wrote many algorithmic composition programs specifically for a single piece. In other words, Ames didn't try to write a single algorithm that could capture many possible compositional approaches. He argued that "such attempts to 'regularize' a decision-making process run the risk of degrading both the quality of the solution and the efficiency of the process in getting to the solution." [14] Rather, Ames wrote individual programs for unique compositional strategies [15].

For example, in *Protocol* Ames employed an exhaustive search strategy that found optimal solutions at each point in the compositional process. In these programs Ames had the computer "evaluate substantial repertories of alternatives in order to select the ones best fitting a protocol of algorithmic tests." [16] This is an example of a strategy Ames called *comparative search*. In the later *Gradient*, Ames incorporated comparative and constrained search. Constrained search "seeks a solution conforming to one or more strict rules, again provided by the composer. Whenever a constrained search encounters some decision in which no option satifies all of these rules, the search backtracks, revises one or more earlier decisions, and tries again." [17] Finally, Ames' Cybernetic Composer program applies "a combination of *constraints* enforcing minimally 'acceptable' standards and *heuristics* encouraging 'preferable' tendencies above and beyond the bottom line." [18]

Most of Ames' techniques are borrowed directly from the computer science literature, but all Ames' algorithms assert the primacy of the programmer-composer. In other words, all Ames' programs operate based on rules provided by the composer, rather than on rules inferred from a data set, or evolved socially. In this sense, Ames' programs, like Xenakis', can only be either combinatorial or exploratory creativity.

Ames' refusal to write general-purpose composition programs [19] might be seen as a refutation of computational creativity. As evidenced by the quote above, Ames seems to think that no algorithm can capture the creative process. Any algorithm, this argument goes, would be inherently limited to the space of compositions laid out by the programmer, and so be unable to truly explore or break out of that space as might occur in human creativity.

4.2. Eduardo Reck Miranda and Artificial Life

Models of artificial life provide an interesting alternative to some of the traditional paradigms of artificial intelligence. In most artificial life simulations, the decision-making process is decentralized, so intelligent behavior results from the interactions of many agents, rather than a simulation of the internal mental processes of a single agent. Peter M. Todd and Eduardo R. Miranda point out how this bottom-up approach might be more appropriate in the domain of music.

> "Creating music is a social activity. Without someone to create it, perform it, and perceive it, music can hardly be said to exist. If we want to build artificial systems that can help us to create music - or, even more, that can attempt to create music on their own - we should strive to include the social element in those systems." [20]

Eduardo Reck Mirando has designed a number of systems where the music is created by multi-agent artificial life systems. His 1992 CAMUS system, for example, sonifies the activities of a cellular automaton. Cellular automata are discrete grids of cells which are connected to their neighbors. Each cell can take on a particular state based on a global set of production rules. For instance, a cell might switch to the *on* state if two of its neighbors are on, or it might switch to the *off* state if none of its neighbors are on. The most well-known cellular automata rules are called The Game of Life, and they describe a system that often seems to come alive with continuous action.

In CAMUS, Miranda sonifies each cell in a 2-dimensional cellular automata to three pitches.

> "When the algorithm produces a live cell, its coordinates are taken to encode the triple of notes starting from a given lowest reference note. For example, if a cell at the position (19, 7) is alive, its coordinates descibe the intervals of a truple of notes: a fundamental pitch is given (the user can specify a list of pitches to be picked by the system), the next note is 19 semitones higher and the last note is a total of 26 semitones above the fundamental." [21]

Rhythms are then chosen based on the distance between consecutive live cells. CAMUS is a perfect example of Boden's combinational creativity. The system isn't exploring or transforming a conceptual space, it merely combines melodic ideas based on rules that are preset by the programmer. In other words, the music that results is primarily supplied by the mapping of the cellular automata to pitches and rhythms based on rules that are set up by the programmer or user. The only exploratory aspect of CAMUS involves the randomized starting state, which in turn leads to different outputs each time the system is started.

More recently, Miranda has created more complex social musicians. In Miranda's 2002 *mimetic model*, for instance, the goal was not necessarily to simulate the composition of music as it is accomplished by an individual composer, but to simulate the way that music arose in human society. In other words, Miranda created agents with abilities and objectives that might lead to the development of a learned repertoire of sounds. The agents were designed to want to be social. Each agent's ability to socialize is based on their being able to understand and imitate the sounds that are played by other agents. Towards this end, the agents have three basic abilities: the ability to play sounds based on a synthesis model that is shared by all, the ability to listen to other sounds, and the ability to imitate what is heard. It should be noted that the agents cannot inspect the memory of other agents, so they must imitate based on observed phenomena; they can see some of the synthesis apparatus belonging to other agents, and they can hear the resulting sound. Miranda describes a typical interaction in his system.

> "The interactions fundamentally occur on a two-by-two basis: one of the agents

> (agent A) plays a sound to the other agent (agent B). Agent B then attempts to imitate agent A. Note that the agents can only understand what they hear in terms of the sounds that they already know. Therefore, agent B must play back a sound taken from its repertoire. If its repertoire is empty, then it makes a response at random and plays it. If its repertoire is not empty, then it carefully listens to agent A and selects the most similar sound that it can find in its repertoire for the imitation. Next, agent A assesses whether the imitation is acceptable and gives a feedback to agent B. If the imitation is acceptable, then agent A reproduces the original sound as a positive feedback flag. However, if it was unacceptale, the agent A remains quiet and silently terminates the interaction, signaling agent B that its imitation was a failure. After this interaction, both agents inspect their memories and perform a few updates based upon the experiences they both had." [22]

After around 5000 of these types of interactions, a community of agents has been shown to build up a repertoire of eight to twelve sounds. Miranda is clear to note that these agents aren't really generating music. At best, they're very crude sound designers with multi-faceted abilities that might reflect human social interactions at some basic level. Nevertheless, the system might be thought of as creative because it is generating sounds without relying on maps created by the programmer. In this sense, this system can be classified as implementing exploratory creativity and possibly transformational creativity.

Genetic algorithms are another major paradigm of artificial life that has been profitably explored by algorithmic composers. Bruce L. Jacob's *Variations* system is interesting in part because he applies genetic algorithms to the composition algorithm rather than the musical material itself.

In genetic algorithms, concepts borrowed from Darwin's theory of evolution are simulated on populations of data. The phenotypes, or agents, are first created randomly, then allowed to reproduce with one another, combining or mutating the underlying data. A fitness function is used to decide which individuals may reproduce and which may not. As long as the fitness function has some bearing on the desired result, genetic algorithms will evolve ever closer to an acceptable solution.

Jacob's *Variations* system has three components, the composer, the ear and the arranger. The composer system generates musical material, while the ear approves or disapproves it based on its own set of criteria, and the arranger orders the output into something it deems musical. The rules that are employed by each component are the phenotypes. The rules are randomly generated at first, but a human listener rates each rule in each generation, and they are allowed to reproduce accordingly. Only after many generations is allowable musical material finally generated which corresponds to the human's aesthetic preferences and the software's evolved rule systems.

Jacob's system is significant because he argues that it closely models his own creative processes [23]. His own process is one of refinement and variation, where he sets up processes for a specific piece, then generates melodic variations using those processes.

Jacob's work is difficult to classify because of a lack of specificity in his publications. *Variations* might be seen as implementing transformational creativity, since the evolving rule systems could combine to form systems that generate music that was outside of the composer's own specification. This depends on how the data structures in which the rules are represented, and the multiplicity of ways in which they can be transformed.

4.3. David Cope and Learning Systems

David Cope has implemented many algorithmic composition systems, but in this essay I am going to focus on one of his more recent and most successful systems, *Emily Howell*. This system is interesting in part because it is Cope's attempt to model human creative processes, and he takes some inspiration from Boden's conception of creativity.

It might be useful to revisit Boden's work at this point. We can summarize Boden's framework with three basic axioms.

1. Creativity can be personally creative (P-creative) or historically creative (H-creative).
2. Creativity can be combinational, exploratory or transformational, depending on how a system interacts with conceptual spaces.
3. Creative algorithms must produce something that is judged to be valuable, interesting or useful in some way.

Finally, we must keep in mind Boden's somewhat vague caveat that, for a system to be truly creative, it must have some agency in traversing or transforming a conceptual space. A program cannot be following a set of pre-programmed rules. It is useful to reiterate here that Boden's criteria might be more useful if also accompanied by an empirical anlysis of the output, as suggested by Ritchie, but executing such analyses is beyond my means for this article.

Many of Cope's programs are built on the fundamental principle that rules shouldn't be written into a program, but learned by that program based on a corpus of input data. This applies to his famous *EMI* program from the late 1980s, which succeeded in using recombinance to generate style imitations of long-dead composers, and to *Emily Howell* as well. *Emily Howell* is built on an underlying data structure called an *association network*. Association networks are complete, weighted, bi-directional networks of symbols. In other words, they're a database of symbols where every symbol has a weighted connection to every other symbol. The symbols might be letters, or colors or music data.

The association networks employed by *Emily Howell* are generated from a collection of single measures of music that are derived from a corpus of input data. The weightings are generated primarily based on proximity. In other words, events that are closer together are weighted more strongly. Cope emphatically notes that there is no predispositional weighting scheme built into *Emily Howell*. In other words, it is not giving extra priority to triads or the rhythms of natural phrases. It should be noted, however, that when interpreting an association network, the program does understand the musical concepts of transposition and voice reordering [24]. Unlike some other similar networks, such as markov models, association networks generate deterministic, non-probabilistic weightings. As the database is being built, the weightings "constitute a kind of dynamic rules system that changes over time with new input." [25] The association network itself does not correspond to a creative mechanism. After a network is built, Cope applies an *inductive association* process to simulate creative thought.

Inductive association is a way for the program to choose the next output symbol, based on the previous output symbol, without making the obvious choice of the hightest weighted symbol. Inductive association

> "informs the association network that instead of limiting itself to obvious deductive associations, it should incorporate reasonable substitutes. 'Reasonable' here means that the program carefully sifts through the associated weightings of possible words other than the highest-weighted words when forming its replies, and chooses one of those that is less likely than its deductive choice but nonetheless ligitimately competes for the highest weighting." [26]

In essence, inductive association, as Cope defines it, means that when the program chooses its next output symbol, it considers not only the direct weights from one symbol to the next, but also the weights from the neighborhood of the current symbol to the neighborhood of the symbol under consideration.

Cope then defines three layers of inductive processes that correspond to Boden's three types of creativity: combinational, exploratory and transformational. In what Cope calls *local creativity*, a single symbol is selected in the procedure described above. This corresponds to Boden's combinational creativity, and allows the system to combine familiar, related symbols.

Cope defines *regional creativity* as an allusional process, whereby a whole string of symbols is grabbed from another portion of the networked and transformed using transposition and revoicing. This corresponds to Boden's concept of exploratory creativity, as the program eseentially transplants a string of symbols from one part of the network to another.

Finally, Cope also implements what he calls *global creativity*, which corresponds to Boden's transformational creativity. Conceptually, there is no real difference between Boden's version of the concept and Cope's. Rather than just grabbing an allusion from another part of the network, the program actually connects the current symbol to a symbol that is in a completely different and poorly related neighborhood. It strongly weights a previously poorly weighted connection.

> "Another way to look at these three types of musical creativity would be that local induction causes simple rethinkings of old material and processes, regional induction involved one-time appropriation of other music, and global induction allows an influence from the musical environment to cause slight but fundamental shifts in musical style to occur." [27]

In theory, Cope's *Emily Howell* can accomplish all of Boden's forms of creativity. This ideological agreement should be expected though, since Cope was fully aware of Boden's book when he wrote the program, and certainly partly influenced by Boden's work; however, there are a few significant caveats that should be noted.

First, *Emily Howell*'s database is filtered by the user. This is not to say that the program employs supervised learning. Although versions of the program can be supervised, the final form of the program that Cope has used to generate musical output is fully autonomous. Rather, it's significant that the input fed to the program isn't just all music ever written. In fact, this caveat applies to all learning music systems of which I am aware. In future work, it might be interesting to allow a program like *Emily Howell* to learn by simply crawling the internet, without distinguishing good MIDI files from bad, or even necessarily indicating what is music and what isn't. That might be the only practical way to assert that the programmer has not filtered the input set.

Similarly, it should be noted that Cope has also filtered *Howell's* output. In other words, he picks the best of her output to present to the world. This is a fully reasonable expectation. No human composer shows all of their output to the world. All human composers filter out whatever they perceive as mistakes. Still, before conferring true creativity on a machine, it's important to ackowledge the front-end and back-end filtering that is, by necessity, implemented by the programmer.

Finally, it should be noted that *Emily Howell* only has a single context of data. It only understands musical notes, or text, or synthesis events, or whatever the database is set to handle [28]. This limitation in itself might imply that *Emily Howell* is really only achieving exploratory creativity. The program can never, for instance, hear the similarity between inharmonic FM synthesis and bell sounds, then insert a reference to John Chowning in a piece for tubular bells.

5. Conclusions: Are Computers Creative? Are People Creative?

In this article, I've shown many different systems for algorithmic composition. All of them have been successes in academia, and some of them have achieved commercial success through software distribution or sales of musical output. Yet, all of these systems employ algorithms that were almost entirely pioneered by earlier AI researchers. The primary difference between these systems and their AI counterparts is the inclusion of some sort of non-deterministic element. In other words, chance is incorporated at some level, with the exception of some of the programs by Charles Ames. So if creative methods are the same as the methods for general problem solving, why do we classify creativity as something else?

It turns out that not all scholars do see creativity as something different.

> "Back in 1958, Alan Newell, J.C. Shaw, and Herbert Simon circulated a monograph entitled *The Processes of Creative Thinking*. In it they asserted that: 'creative activity appears simply to be a special class of problem-solving activity characterized by novelty, unconventionality, persistence, and difficulty in problem formulation.'" [29]

In my opinion, creativity is a linguistic device created by humans to apply a positive value judgement to a successful or otherwise valuable instance of problem solving. When we say that a work is *creative* we are simply saying that we think it is good, or at least better than some of the alternatives in some way. Similarly, when we say a program is creative, we are simply conferring exceptional status to that program.

Composers and artists who search for *creativity* are essentially studying problem solving in the musical domain. The most successful algorithmic artists, such as Cope, Miranda and Ames, have been successful in part by adapting techniques which were previously successful in the domain of artificial intelligence. This should not be considered a denigration of their work or a negation of their contributions to the arts in general. Rather, this is simply an acknowledgement that algorithmic art is branch of artificial intelligence with a unique and difficult problem.

Notes

1. Specifically, I will limit this review to a few composers of *computer-assisted* algorithmic music.
2. See Cope 2005 for a large list.
3. Margaret A. Boden, *The Creative Mind: Myths and Mechanisms*, 2nd ed. (London: Routledge, 2004): 2.
4. Ibid., 3.
5. Ibid., 58.
6. Ibid., 6.
7. Ibid., 41.
8. Ibid., 148.
9. Graeme Ritchie, "Some Empirical Criteria for Attributing Creativity to a Computer Program," *Minds & Machines* 17 (2007): 70.
10. Ibid., 72-73.
11. Ibid., 90.
12. Iannis Xenakis, *Formalized Music: Thought nd Mathematics in Composition* (Bloomington, IN: Indiana University Press, 1971): 132-133.
13. Margaret A. Boden, *The Creative Mind: Myths and Mechanisms*, 2nd ed. (London: Routledge, 2004): 121.
14. Charles Ames, "Quantifying Musical Merit," *Interface* 21, no. 1 (1992): 71.
15. Ames did create a program for composing music using markov models, and he theorized specifically about general strategies for quantifying merit in algorithmic composition.
16. Charles Ames, "Protocol: Motivation, Design and Production of a Composition for Solo Piano," *Interface* 11, no. 4 (1982): 213.
17. Charles Ames, "Stylistic Automata in Gradient," *Computer Music Journal* 7, no. 4 (1983): 45.
18. Charles Ames, "Quantifying Musical Merit," *Interface* 21, no. 1 (1992): 54.
19. Although the Cybernetic Composer wrote music in multiple popular genres, it only worked in a subset of music composition that incorporated strictly defined instrumental roles, such as using the piano to play the chord progression, while the melody is carried by a melody instrument.
20. Peter M. Todd and Eduardo R. Miranda, "Searching for Computational Creativity," in *Musical Creativity: Multidisciplinary Research in Theory and Practice*, edited by Irene Deliege and Geraint A. Wiggins (New York: Psychology Press, 2006), 376.
21. Ibid., 380.

22. Eduardo Reck Miranda, "Emergent Sound Repertoires in Virtual Societies," *Computer Music Journal* 26, no. 2 (2002): 80.
23. See Jacob, 1996.
24. In other words, if the input database contains a root-position C-major triad moving to a root position C#-diminished triad, it will understand that as equivalent to a 1st-inversion G-major triad moving to a 1st-inversion G#-diminished triad. These rules are not preset weights or pre-programmed rules, they are more like the basic mathematical operators of common-practice-notation music.
25. David Cope, *Computer Models of Musical Creativity* (Cambridge, MA: The MIT Press, 2005), 282.
26. Ibid., 289.
27. Ibid., 316.
28. It might be the case that there is a version of *Emily Howell* that can work with sets of differently-parameterized data; however, this is not clearly indicated in the literature, nor has Cope indicated this in personal communication.
29. Charles Ames, "Quantifying Musical Merit," *Interface* 21, no. 1 (1992), 53-54, quoting Newell, Alan, J.C. Shaw and H.A. Simon, *The Process of Creative Thinking* (Santa Monica, CA: RAND Corporation, 1958), 5.

Bibliography

Ames, Charles and Michael Domino. "Cybernetic Composer: An Overview." In *Understanding Music with AI: Perspectives on Music Cognition*, edited by Mira Balaban, Kemal Ebcioglu and Otto Laske, 187-205. Cambridge, MA: The AAAI Press, The MIT Press, 1992.

Ames, Charles. "Quantifying Musical Merit." *Interface* 21, no. 1 (1992): 53-93.

Ames, Charles. "Protocol: Motivation, Design and Production of a Composition for Solo Piano." *Interface* 11, no. 4 (1982): 213-238.

Ames, Charles. "Stylistic Automata in 'Gradient'." *Computer Music Journal* 7, no. 4 (1983): 45-56.

Balaban, Mira, Kemal Ebcioglu and Otto Laske, ed. *Understanding Music with AI: Perspectives on Music Cognition*. Cambridge, MA: The AAAI Press, The MIT Press, 1992.

Boden, Margaret A. *The Creative Mind: Myths and Mechanisms*. 2nd ed. London: Routledge, 2004.

Cope, David. *Computer Models of Musical Creativity*. Cambridge, MA: The MIT Press, 2005.

Dean, Roger T., ed. *The Oxford Handbook of Computer Music*. Oxford: Oxford University Press, 2009.

Jacob, Bruce L. "Algorithmic composition as a model of creativity." *Organised Sound* 1, no. 3 (1996): 157-165.

Jacob, Bruce L. "Composing with Genetic Algorithms." Paper presented at the International Computer Music Conference, Banff, Alberta, Canada, September 1995.

Miranda, Eduardo Reck. "Emergent Sound Repertoires in Virtual Societies." *Computer Music Journal* 26, no. 2 (2002): 77-90.

Nierhaus, Gerhard. *Algorithmic Composition: Paradigms of Automated Music Generation*. Wien: Springer-Verlag, 2010.

Ritchie, Graeme. "Some Empirical Criteria for Attributing Creativity to a Computer Program." *Minds & Machines* 17 (2007): 67-99.

Todd, Peter M. and Eduardo R. Miranda. "Putting some (artificial) life into models of musical creativity." In *Musical Creativity: Multidisciplinary Research in Theory and Practice*, edited by Irene Deliege and Geraint A. Wiggins, 376-395. New York: Psychology Press, 2006.

Wiggins, Geraint A. "Searching for Computational Creativity." *New Generation Computing* 24, no. 3 (2006): 209-222.

Xenakis, Iannis. *Formalized Music: Thought and Mathematics in Composition*. Bloomington, IN: Indiana University Press, 1971.

Evan X. Merz

Evan Merz (b. 1981) is a composer, programmer and blogger based in San Jose. He obtained a Bachelor's Degree in computer science from the University of Rochester in 2004, and a Master's Degree in computer music from Northern Illinois University in 2010. His music has been performed at the CSUF New Music Festival 2013, April in Santa Cruz 2013, Currents Santa Fe 2012, University of South Dakota 60/60 2012, Phono Photo No. 6, Silence, Beauty and Horror 2009, musicBYTES 2009 and New Music Hartford 2009. Evan is the author of Sonifying Processing: The Beads Tutorial, which introduces sound art to Processing programmers. He also works heavily as a freelance composer, scoring for numerous videogames and television productions. He is the SEAMUS Webmaster and the blogger at computermusicblog.com. Currently, Evan is a DMA candidate in University of California at Santa Cruz algorithmic composition program.

Introductory Text

This paper reviews the algorithmic composition software by three groups of composers, and shows how their work relates to the model of creativity proposed by Margaret Boden.