Abstract

The chapter describes the manifold interactions between composers and computers in the creative process. This historically grown dialogue has meanwhile transcended the paradigm of the traditional compositional process and leads to a number of interesting implications and side effects in the context of current trends in algorithmic composition, generative music and computational creativity. Not only the computer per se, but especially the choice of a certain software and already a specific view on the musical material turns into a decisive factor in, often interlinked, generative and analytical approaches–whereby strategies of musical representation and the choice of a particular mapping are of crucial importance. Finally, it is examined how creativity can be defined and located in this creative interaction between man and machine.

Introduction

The relationship between computer and composer is characterized by a dense network of creative interactions that cannot simply be reduced to a hierarchical order of instruction and execution. There is no need to invoke elaborate algorithms of artificial intelligence to address the computer's role as a plain utensil for realizing compositional strategies. Choosing a specific tool, but also just working within self-imposed or given constraints, in itself produces side effects that retroact on the compositional process. Broadly speaking, compositional perspectives condition systematic approaches, which in turn continue to influence the perspectives–and the computer is instrumental in this interplay. There are plenty of paradigmatic examples for this, which can be cast as dichotomies: Time-based versus hierarchical approaches to algorithmic composition; knowledge-based versus non-knowledge-based systems for musical analysis; even graphical versus textual computer music languages each suggesting different approaches in the compositional process.

No matter how the computer is used in the creative process, there is usually one commonality: From the required algorithmic description arises a model, abstracted from an individual case, and consequently a meta-system for the generation or transformation of an entire class of musical structures, whether these are sounds or symbolic data.

This property of abstraction alone reveals a difference to the traditional compositional process: Working with the computer, the composer faces not only a single musical instance as their counterpart, but in fact a system with which they may interact with and reflect on in different ways and whose compositional premises and generated structures may at all times be scrutinized and amended.

As the following sections look more closely at the distinct approaches to creative work that uses the computer, it will also become evident that the diversity of procedures and aesthetical implications leads to an unavoidable vagueness in any delineation of categories. Nonetheless, as a form of creative boundary-crossing, this vagueness may itself become a crucial element of the compositional process.

Genesis

In 1936, Alan Mathison Turing defined algorithm and computability as formal mathematical terms, yielding an abstract device called the Turing machine (Turing 1937). In theoretical computer science, this model of an automaton permits the computation of any algorithm, given that it is computable at all. The Turing machine defines a class of compatible functions whose way of processing served as a suggestion for the development of modern imperative programming languages. An equivalent model for computability had been in development already since 1932 by Alonzo Church. His lambda calculus became essential for the advance of functional programming languages. Finally, in 1945, John von Neumann's First draft of a report on the EDVAC (Neumann et al. 1945) delivered a concrete system design for building a computer aimed at (theoretically) universal problem-solving. This report consists of theoretical reflections on the construction of a digital computer, assembling the ideas of von Neumann, but also of J. Presper Eckert, John Mauchly, and Herman Goldstine.

As early as 1942, Konrad Zuse developed Plankalkül (plan calculus), a first draft for a high-level programming language, inspired by the lambda calculus. Plankalkül included pioneering constructs such as assignments, function calls, conditionals, loops and compound data types. Although this programming language had not actually been implemented at the time due to technical limitations, the concept was proven valid by an implementation in 2000 (Rojas et al. 2004).

The first programming languages to become relevant in a musical context originated in the United States in the 1950s. In 1956, the first Dartmouth Conference was held on the topic of artificial intelligence, a new discipline whose name was coined by John McCarthy, also known as the author of Lisp. This functional programming language was developed in 1958 at MIT and based on the lambda calculus. Its imperative counterpart Fortran had been conceived even earlier in 1953 by John W Backus, and was realized the following year at IBM.

Lisp would become one of the key programming languages of artificial intelligence and also later, along with numerous dialects, an important language for systems of algorithmic composition. These include "FORMES" (Rodet and Cointe 1984) the "Crime environment" (Assayag, Castellengo and Malherbe 1985) and "Patchwork" (Laurson 1996), which can be seen as the predecessor of "OpenMusic" (Agon, Delerue, and Rueda 1998). While these systems had been developed at the IRCAM, in 1989, during a residency at Stanford, Rick Taube began the development of "Common Music" (Taube 1991; Taube 2004), another Lisp-based programming environment for algorithmic composition still in use today.[1]

The use of Fortran for musical applications had been pursued most notably by Iannis Xenakis[2], who in 1961 gained access to an IBM 7090 computer at the Paris branch of IBM. Using his Fortran-based "Stochastic Music Program", he realized the well-known pieces of his "ST series" in the following year. In 1963, Lejaren Hiller and Robert Baker created the Fortran-based "MUSICOMP" (Music Simulator Interpreter for Compositional Procedures), the first software for the simulation of musical composition procedures.

Besides the development of the first high-level programming languages, the 1950s and early 1960s also gave birth to a series of influential and pioneering advancements in computer music. In 1957, while Edgard Varèse and Iannis Xenakis were working on their compositions for the Philips Pavilion at the Brussels World's Fair 1958, Lejaren Hiller and Leonard Isaacson implemented the "Illiac Suite" at the University of Illinois. This string quartet is commonly regarded as the first algorithmic composition created by a computer, although in the same year a computer at Harvard was employed by Frederick Brooks to analyze and synthesize common-meter hymn tunes using Markov models up to order 8 (Brooks et al. 1957). Markov models for stochastic composition had been also suggested in the previous year by Richard Pinkerton in an article for the Scientific American detailing a "banal tunemaker" (Pinkerton 1956). Following Hiller, it was mainly Pierre Barbaud[3] and Roger Blanchard who, in France from 1959 onwards, developed computer-based algorithmic composition.

Concurrently with the "Illiac Suite", Max Mathews for the first time synthesized a short melody using a computer running the "Music I" program developed at the Bell Laboratories. In the following years, this program evolved into what became known as the "Music N" computer music languages, and led to the design of "Csound"(1986) by Barry Vercoe at the electronic studio of the MIT. "Csound" in turn became the starting-point of computer music languages that could be used both for algorithmic composition and sound processing and synthesis. This included "MAX", developed at the end of the 1980s by Miller Puckette at IRCAM, and in the open source variant "Pure Data" (1996), also written by Puckette, as well as "SuperCollider" (1996) by James McCartney (McCartney 1996).

The development was not only advanced by new software, but also by the creation of new data protocols (MIDI in 1983, OSC in 1997), new interfaces, and, naturally, the increasing capabilities of hardware, opening new perspectives. The ensemble of these factors, but also simply the decision to integrate the computer as a proper medium into the compositional process, led to a number of side effects and far-reaching consequences that had a notable impact on creative work.

Sound and/or Symbol

One commonly distinguishes between the generation of sound and the generation of structure in the use of computers in the compositional process. This coincides with the distinction between sound transformation and sound editing on the one hand, and on the other hand the generation of symbolic values interpreted as musical parameters such as pitch, volume and duration. Although this chapter emphasizes algorithms for the generation of structures over those for the generation of sound, the boundaries between the two domains are fluid. The amalgamation of these poles can be found precisely at the architectural foundation of today's computer music languages addressing both algorithmic composition and sound synthesis. Furthermore, a unifying perspective is deliberately articulated both by dominant historical and contemporary currents:

The first prominent example is Karlheinz Stockhausen in his essay …wie die Zeit vergeht… (…How Time Passes…) (Stockhausen 1956). Here, he proposed a musical perspective in which the domains of form, rhythm and timbre can be seamlessly translated into one another through the new technical means. This idea was condensed in his 1972 lecture "The Four Criteria of Electronic Music":

"There is a very subtle relationship nowadays between form and material. I would even go so far as to say that form and material have to be considered as one and the same … a given material determines its own best form according to its inner nature." (Stockhausen 1989, 111)

Whereas this statement, of course, specifically expresses the creative drive of the Cologne school of electronic music aimed at controlling all musical parameters through serialism, analogous positions can be found earlier in a 1937 lecture by John Cage:

"The composer (organizer of sound) will be faced not only with the entire field of sound but also with the entire field of time. The "frame" or fraction of a second … will probably be the basic unit in the measurement of time. No rhythm will be beyond the composer's reach." (Cage 1937, 5)

For today's composer Horacio Vaggione, the difference between sound and structure becomes almost obsolete:

"I assume that there is no difference of nature between structure and sound materials; we are just confronting different operating levels, corresponding to different time scales to compose." (Roads 2015, 109)

This viewpoint is also shared by Curtis Roads through his "multiscale" conception, "where we can manipulate an entire composition, or its sections, phrases, and individual sounds with equal ease." (Roads 2015, 9)

Composition: Traditional and Computer-based

Irrespective of the specifics of how the computer is used in the compositional process, the mere fact of its use in itself exhibits a number of implications that lead to interesting alternatives and extensions to the traditional process.

In the traditional compositional process, usually a symbolic notation of an imagined sound result is created under particular constraints, destined for one single work. In contrast, a computer-based approach often uses a system of directives to create an entire class of possible compositions, and within this process individual instances may be generated, often with the capacity to render them immediately audible. Of course, this is a generalization, since it is also possible in the conventional context to write, for example, aleatory scores that permit different realizations, just as computer-based contexts permit the use of a deterministic algorithm to produce one fixed musical structure. It is also clear that the possibility of the immediate audible feedback to the generated structures that what entices a procedure of trial-and-error has only become possible through increased computing power, and so earlier interaction models commanded other approaches and aesthetics of the compositional process. On the other hand, composing with traditional means may also be based on auditory feedback, for instance when composing on the piano, a practice for which Igor Stravinsky argued in his autobiography: "It is a thousand times better to compose in direct contact with the physical medium of sound than to work in the abstract medium produced by one's own imaginations." (Stravinsky1936, 5)

If the computer is used to generate sounds or symbols, this is mostly framed by a set of rules that were defined either by the composer or as part of a standard class of algorithm. By contrast, the traditional way of composing is framed by a number of extrinsic constraints, although additional rules as such may be consciously defined or in fact applied unconsciously. Even when relying mostly on their own intuition, the composer is still confronted with various constraints that determine the compositional structure to a certain degree, such as the dynamic, articulatory or pitch capabilities of the instruments, the degrees of freedom inherent to a chosen form of notation, and many more. Apart from these obvious restrictions, constraints can also be understood as internalized compositional premises and restrictions in the traditional setting. In line with this, a convincing definition is given by Vaggione:

“I use the expression “constraint” in the sense of its etymology: limit, condition, force, and, by extension, definition of the degrees of freedom assumed by an actor in a given situation within self-imposed boundaries. In this broader sense, the composer’s constraints are specific assumptions about musical relationships: multi-level assumptions that can be in some cases translated into finite computable functions (algorithms), …” (Vaggione 2001, 57)

Whether the computer is used to mediate someone else’s or to generate one’s own musical structures, in either case a dialogue is initiated that leads to several possibilities of interaction (as well as side effects), distinguishing it from the traditional modus operandi.

Analytical and Generative Approaches

The usage of computers in the musical context could be roughly categorized as either analytically or generatively driven, manifested through a series of different disciplines such as algorithmic composition, sound synthesis, sound analysis, or the modeling of particular musical styles.[4] And yet analytical and generative approaches are often interlinked, especially when working with classes of algorithms that permit both perspectives, such as generative grammars (cf. Laske 1974; Roads and Wieneke 1979; Steedman 1984; Hughes 1991; Chemellier 2004; Rohrmeier 2011) and Markov models (cf. Chai and Vercoe 2001; Allen 2002; Allen and Williams 2005; Schulze and van der Merwe 2011). In these cases, a preceding analysis of one’s own “notated” or improvised music yields a body of rules describing an entire class of compositions, and their evaluation can be said to produce, per definition, stylistic copies as instances of a common structural idea. A paragon for such a twofold application is the computer music software “Bol Processor”. It is based on the formalism of generative grammars and was originally developed in the 1980s for the analysis of tabla music (Kippen and Bel 1989) and later established itself as a system for algorithmic composition (Bel 1998).

Regardless of whether resynthesis by analysis or a “pure” generative approach is chosen, two principle modes of access can be specified. The knowledge-based mode requires problem specific knowledge pertaining to the respective musical domain, like basic rules of tonal harmony or voice-leading. In the non-knowledge-based mode, a rule system is created, for example, autonomously and solely on the basis of musical samples.

The first mode encompasses all those strategies that seek to produce musical structures through a rule-based system whatever its nature. In principle, the output of the system in this category can be largely predicted, and the computer is primarily used for the automation and acceleration of processes that could, in theory, also be carried out with pencil and paper. Examples of this mode are all types of systems in which knowledge about the musical domain is expressed as rules and constraints that are applied in a stochastic or deterministic manner to either generate musical structure or to aid analysis, for instance through style synthesis. Beside a number of prominent techniques and systems for algorithmic composition, one may also assign many historical instances to this category, such as the aforementioned musical dice games and aspects of twelve-tone technique and serialism, all of which allow, technically, both a manual and a computer-aided calculation.

The second mode captures all those strategies where, on the one hand, musical material is autonomously produced, structurally similar to an underlying corpus. This includes Markov models, procedures of grammatical inference (cf. Kohonen 1987; Kohonen 1989; Pachet 1999), or transition networks such as those used by David Cope, who generates compositions in the styles of different genres with his "EMI" (Experiments in Musical Intelligence) system (cf. Cope 1987; Cope 2001; Cope 2014). On the other hand, some non-knowledge-based systems, like genetic algorithms can work without a corpus and may also produce unexpected output (cf. Horner and Goldberg 1991; Biles 1994; Papadopoulos and Wiggins 1998; Gartland-Jones and Copley 2003; Nierhaus 2015, 165-187).

An interesting comparison between algorithms in these two categories is given by Phon-Amnuaisuk and Geraint Wiggins who investigated the effectiveness of a rule-based system in comparison to that of a genetic algorithm, using the scenario of automatic harmonization of a chorus.[5]

Even systems defined by simple rules and completely deterministic behavior may produce unexpected output if the interaction of basic components leads to emergent system behavior and consequently to complex and unpredictable results, the cellular automaton being an example.

Often a distinction is drawn between probabilistic and deterministic systems, whereas in the compositional process there might be a conceptual choice between explicitly preferring chance operations or rejecting them, however, moving on to the level of algorithm classes, drawing this distinction becomes intricate. Leaving aside the fact that a digital computer can only simulate random values through pseudo-randomness,[6] it may be difficult to discern the output of complex systems or deterministic non-linear chaotic algorithms from random behavior. Finally, randomness is an essential component of most generative computer-based procedures, since the implementation of algorithms almost always defines a "stochastic scope" that allows the generation of different instances of a common structural idea.

Algorithmic Composition

The terms algorithmic composition and generative music may be used synonymously or assigned to different categories. Here we distinguish them as follows: "Approaches of algorithmic composition generate a musical structure as the result of a computation, determining an entire composition or parts thereof. Generative music approaches produce the basic conditions for a system that can then evolve autonomously within certain boundaries." The generative approach is also often linked to an act of co-creation, wherein authorship of a composition can no longer be—or is intended to no longer be—attributed to the composer alone.

There are varying definitions of algorithmic composition, something that is already reflected in the distinct but often interchangeably used terms of algorithmic composition and computer assisted composition (CAC) that suggest different scopes for the algorithmically generated structures relevant to the composition: Either a composition is algorithmically determined as a whole, or algorithmic procedures are solely used to generate selected aspects or sections of a composition.

Strategies of algorithmic composition can either be modelled through the "personal strategies" of composers or through a series of common classes of algorithm, e.g. genetic algorithms or Markov models. The former denotes approaches that are highly idiosyncratic in the work of composers and that cannot be simply subsumed under one of these common algorithms. Naturally, "personal strategies" might also be modeled by standardized classes of algorithm, nevertheless it is sensible to maintain the distinction, since the personal choice of a particular strategy or class of algorithm coincides with a particular point of view on the compositional process. The same caveat applies to the theoretical equivalence of certain classes of algorithm: For example, some Lindenmayer-systems may be represented by cellular automata. Despite such interchangeability at the informatics level, deciding to choose either formalism becomes an essential aspect of creative work and often relates to specific aesthetic positions. Some distinctive classes of algorithm almost suggest their status as paradigms of algorithmic composition (cf. Nierhaus 2009, 3-6) by implying both a specific treatment and perspective on the musical material.

Markov models, originating from linguistics, as well as generative grammars are in principle very well fitted for the processing of one-dimensional context-based sequences of symbols. On the downside, they are not very well fitted to account for dependencies between horizontal and vertical musical features (or, in general, across multiple dimensions). In contrast, neural networks were originally developed for image processing and classification. Here, the processing of a temporally evolving context conditions the modification of network topologies and the search for suitable representations of musical information. However, the way a generative system treats time is not only reflected in its manipulation of musical input data but also in the mode of data output. Thus, a generative grammar typically produces its terminals only at the very end of all substitutions, whereas a genetic algorithm continuously emits data until the process is interrupted or the fitness criteria have been met, while cellular automata mutate their cell states in cycles of varying length, lacking a temporal limit or target value.

Therefore, the temporal flow of music is not necessarily reflected in the workings of the generative algorithms. Their underlying concepts are either process-based, emitting a constant stream of musical information, or goal-oriented, delivering the solution to a specific task. Apart from the treatment of time, different classes of algorithm also take very different stances towards the structural meaning of the analyzed or generated material: Heinrich Schenker's assumption of an imaginary Ursatz (fundamental structure) that leads to a composition through a multi-layered Auskomponierung(composing-out), or Fred Lerdahl's and Ray Jackendoff's Generative Theory of Tonal Music (Lerdahl and Jackendoff 2010) where a generative view leads to an extensive model of representing tonal music; these and similar approaches are of course best modeled by a generative grammar, adopting a hierarchical treatment of the musical material. An opposing view is reflected in the information processing of a cellular automaton, in which cells update their states according to their internal state and the states of neighboring cells. Consequently, these and other properties of algorithm classes determine not only the modus operandi of generating and transforming musical information, but also specific directions within the compositional process, suggesting particular aesthetical positions.

Finally, and additionally, encoding, representation and, in particular, the overall mapping strategy are of utmost importance, since they define the interface between information processing and musical structure. There are varying interpretations of the terms “encoding” and “representation.”[7] In order to describe their respective implications, we use the following definition: “An encoding converts musical information into a format suitable for the internal processing of an algorithm, whereas a representation displays the musical material from one or several viewpoints.”[8]

The type of encoding of musical information may have distinct effects on the compositional process. Whether, for example, the architecture of the computer music system uses a binary or a decimal representation of values would commonly have little influence on the musical structure generated. On the other hand, encoding musical information through data protocols such as XML (Extensible Markup Language) (cf. Steyn 2013), MIDI (cf. Roads 1996, 969–1016) or OSC (Open Sound Control) (cf. Wright 2005) evokes a number of diverging possibilities and constraints for processing and representing musical structures. An even greater influence is exerted if the choice of encoding implicates the pivotal manner of processing information, as is the case with some types of data compression used in algorithm composition.[9]

With regard to representation, a descriptive example is the seemingly trivial choice of interpreting numbers either as intervals or absolute pitches. This choice leads to different levels of abstraction, but also yields different degrees of error susceptibility.[10] Moreover, multi-dimensional representations allow for a more nuanced view on the musical material, since a parameter can be presented from different angles. A presentation of this kind can be found in a generative system by Michael Mozer (Mozer 1994), who fed a neural network with a multi-dimensional representation for the pitch parameter according to a model by Roger Shepard (Shepard 1989). [11]

Ultimately, of critical importance is the overall mapping strategy, i.e. the specific notion of how the properties of a generative system shall become manifest in the musical structure. Of course, in the interplay of algorithm and musical output, all those solutions that lead to a satisfying compositional result may seem valid. If, however, the objective is to adequately reflect the specificities of an class of algorithm within the musical structure, one might want to avoid those strategies that either disregard the particular algorithm’s system behavior, or that use “a mapping of a mapping” in the procedure of musical translation. An example is the assignment of a pitch scale to the “raster coordinates” of the cells in a cellular automaton. Giving meaning to these coordinates obliterates the automaton’s system behavior which exclusively depends on the internal state of a cell and of its neighboring cells, irrespective of its position in a matrix. Nevertheless, this pitch mapping is very common and was already used in some of the early approaches of using cellular automatons.[12] An inventive way of producing polyrhythmic structures, based on location-dependent state changes of the cells of a Boolean network (a network related to CA), was described by Alan Dorin (Dorin 2000). Further problematic situations arise if the particular and sometimes complex behavior of an algorithm class is reduced to the production of more or less random values (Beyls 1989; Beyls 1991; Millen 1990; Hunt and Orton 1991), or if its system behavior is rendered unrecognizable by mapping a structure that is the result of a prior mapping into a different domain.[13]

Generative Music

In the discussion of the term generative music, highly diverse definitions can be found, chiefly originating from the categorizations of generative art which will be used as the umbrella domain whose definitions then analogously apply to music. In their much-quoted paper "What is Generative Art" (Boden and Edmonds 2010), Margaret Boden and Ernest Edmonds locate a number of currents, mostly beginning in the 1950s, within eleven categories.[14] While this taxonomy reveals several interesting examples, multiple overlaps between the categories as well as their unclear hierarchical order render them largely inutile for the precise identification of the domain of generative art and generative music.

Philip Galanter offers a division into "Simple Highly Ordered Systems"—such as tessellation, numerical series, the golden section and Fibonacci series. "Simple Highly Disordered Systems"—including chance operations and probability theory, noise and random number generators and "Complex Systems"—e.g. fractals, Lindenmayer-systems, neural networks and cellular automata. Organized around classes of algorithms, this division does seem useful for a taxonomy of the various approaches of both algorithmic composition and generative music. But more importantly, his widely quoted general definition of generative art, allows a specification of the domain of generative music:

"Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art." (Galanter 2003, 4)

An even stricter definition is given by Matt Pearson, who adds to the criterion of autonomy, as noted by Galanter, those of unpredictability and collaboration:

"To be able to call a methodology generative, our first hard-and-fast rule needs to be that autonomy must be involved. The artist creates ground rules and formulae, usually including random or semi-random elements, and then kicks off an autonomous process to create the artwork … The second hard-and-fast rule therefore is there must be a degree of unpredictability. It must be possible for the artist to be as surprised by the outcome as anyone else … Creating a generative artwork is always a collaboration, even if the artist works alone." (Pearson 2011, 6)

Both Galanter's and Pearson's criteria are very useful for a definition of strategies of generative music that highlights the specific embedding within a compositional context, regardless of a particular algorithm class used.

Generative music could also be paraphrased more poetically, as Allen Watts did in Nature, Man, and Woman; here, he compared the aesthetics of creation from the perspective of Christianity and Taoism, respectively:

"For from the standpoint of Taoist philosophy natural forms are not made but grown … But things which grow shape themselves from within outwards. They are not assemblages of originally distinct parts; they partition themselves, elaborating their own structure from the whole to the parts, from the simple to the complex." (Watts 1970, 36)

Side-Effects

The particularities of algorithm classes, the design of computer music languages, the strategies of encoding, representing and mapping: all of these demonstrate that within the creative process the computer cannot be reduced to a tool for the impartial execution of instructions. Instead, it induces a dialogic process resulting in a series of undesired but also desired side-effects. The necessary translation of compositional and analytical strategies into a computational context constitutes anything but a "neutral" communication channel, as is of course also the case with the feedback of the medium, whatever its nature. In this, information is not transmitted but transformed, and also eminently determined by the design[15] of the computer music languages employed.

In systems with graphical interfaces such as "MAX", "Pure Data", or "Open Music", objects are "wired" with each other to form a virtual signal flow, thus suggesting a transformative process—also for the generation of music—with a temporally directed motion from A to B. On the other hand, text-based systems such as "Common Music" or "SuperCollider" suggest a different perspective on the musical material in which, for example, functions can be passed as arguments to other function calls, depicting the compositional process rather as the construction of a dense texture of musically influencing constituents.

Besides the software, the quality of the hardware and the interface[16] for its control may also induce different aesthetical positions and compositional approaches. For example, in the history of computer music, the increasing effectiveness of hardware allowed the transition from batch-mode[17] to real-time interaction. These are two approaches that, based on technical givens, suggest aesthetically diverging positions. In the first case, musical results of a generative process are solely amended by editing the algorithms, while in the second case a liberal modification of the system's output is promoted. The first position was for example preferred by Hiller and Barbaud and today is often questioned if not rejected with reference to composers such as Xenakis in favour of the free manipulation of algorithmically generated results (cf. Roads 2015, 348). Another popular but also overstated argument quotes Debussy: "Works of art make rules but rules do not make works of art." (quoted from Paynter 1992, 590). This statement surely possesses validity in the light of rules and constraints, as determined from ex-post-facto analysis of mostly historical musical genres, however it must be questioned in the case of rules established by an artist for the creation of their artworks. Numerous examples can also be found in extra-musical domains, such as in the works of A. Michael Noll, Frieder Nake, Georg Nees and other pioneers of generative computer graphics, proponents of cybernetic art like Nicolas Schöffer and Gordon Pask, but also the exponents of conceptual art, wherein Sol LeWitt's dictum "the idea becomes a machine that makes the art." (LeWitt 1967, 1) likewise aptly describes the formation of works in the context of musical-generative approaches. Curtis Roads advocates free intervention into algorithmically generated structures, arguing that "Computer programs are nothing more than human decisions in coded form. Why should a decision that is coded in a program be more important than a decision that is not coded?" (Roads 2015, 348). One can hardly disagree with Roads' position, since both coded and non-coded decisions—the latter including the entire domain of traditional composition—may produce satisfying musical results. So the two variants can be seen as equivalent with respect to the work, yet there is an essential difference regarding the conditions of production: As opposed to a non-coded decision, the coded decision presupposes awareness of one's own musical constraints, so that these may be formalized on a meta level. It is this property that makes the computer an instrument for the objectification and reflection of compositional strategies, thus opening up new perspectives for musical self-reflection and analysis.

An analysis in this sense was also a crucial aspect of the artistic research project "Patterns of Intuition"(Nierhaus 2015), whose aim was to make visible and objectify unconscious compositional strategies through a cyclical method of personal evaluation and computer-based modeling. Generally, the procedure was thus: Presentation of a compositional principle. Formalization of the approach and implementation in the form of a computer program. Computer generation of musical material. Evaluation of the results by the composer. Modification of the strategy of formalization with respect to the identified objections. Entry into new and further cycles of generation and evaluation until correlation between the computer-generated results and the composers' aesthetic preferences is sufficiently high, or the limits of formalization have been reached. This project did not aim to address musical intuition as a whole and in completely formalizable terms; rather, its goal was to shed light on those particular aspects of intuitively made decisions that can be related back to implicit rules or constraints applied by the composer.

(Computational?) Creativity

The previously discussed implications of using computers are also reflected in an interesting discourse on co-creation, authorship and the question of the uniqueness of digital artworks (cf. Galanter 2012, as well as Ward and Cox 1999). Unavoidably, addressing such issues in a computational context, leads to the question as to whether the process of creation, manifesting itself in various ways of interaction with the computer, can also be autonomously accomplished by the machine itself. Based on current research on artificial intelligence and creativity, the discussion of this question cannot be answered unambiguously, as the relevant terms are not conclusively defined due to lack of general agreement, and often they also integrate clauses that are vaguely or self-referentially formulated. For instance, intelligence is stated to be "… the ability to solve problems, or to create products, that are valued within one or more cultural settings." (Gardner 2011, xxviii), "...goal directed adaptive behavior." (Sternberg and Salter 1982, 24-25),"...the aggregate or global capacity of the individual to act purposefully, to think rationally, and to deal with effectivity with his environment." (Wechsler 1958, 7)–or simply and recursively stated: "...the capacity to do well in an intelligence test" (Boring 1923, 35). The domain of artificial intelligence is by no means more insightful: "Achieving complex goals in complex environments." (Goertzel 2006, 15), or: "Artificial Intelligence (AI) is the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior …" (Barr and Feigenbaum 1981, 3). Shane and Hutter (2006) offer no fewer than 71 notions of intelligence. Definitions of creativity (c.f. Sternberg 1999) come with the same issues: Margaret Boden has offered that "Creativity is the ability to come up with ideas or artefacts that are new, surprising, and valuable." (Boden 2003, 1). Rob Pope unfolds his notion of creativity through "...an extended meditation upon a single sentence: Creativity is extra/ordinary, original and fitting, full-filling, in(ter)ventive, co-operative, un/conscious, fe<>male, re … creation." (Pope 2005, 52).

It is also possible to approach this theme ex-negativo, as Roads states: "Computers are excellent at enumerating possible solutions to a given set of constraints. This has led to optimal performance in well-defined tasks such as playing checkers, where the game has been entirely solved,…" and later: "Music composition is not a finite-state game; it has no simple well-defined solution like checkmate." (Roads 2015, 354-55).

Obviously, the only common denominator of these definitions and notions is their diversity, and it is just this diversity that may resolve the dilemma. Mihaly Csikszentmihalyi presents creativity as a product of social interaction and aesthetic judgments:

"What we call creativity is a phenomenon that is constructed through an interaction between producers and audience. Creativity is not the product of single individuals, but of social systems making judgements about individuals' products." (Csikszentmihalyi 1999, 313).

From here, an analogy can be drawn to compositional work with the machine: "In the interaction with the computer, creativity appears not necessarily as an inherent quality of a generated structure, but rather through an assessment with respect to its musical potential."

This open notion may also be specified by a closer look at the subject and the object of the assessment process by splitting up the "social systems" (making judgements) and the "individuals' products" (being judged) assuming that judgments are made by humans and computers are being judged. Let's further specify "individuals' products" as artefacts produced by a program, and a "social system" as a compound of a user, who might also be the programmer and the audience or the society to which the computer-generated artifacts become relevant.

To deal with computer music (herein understood as an umbrella domain for sound synthesis, algorithmic composition and generative music) within these categories it is useful to shift the perspective to a meta-musical domain, namely computational creativity which brings various artistic and scientific approaches into play–firstly because most algorithms of musical structure genesis are not specific to music (a poetry generator as well as a melody-maker may both be based on a Markov model). Secondly because computational creativity provides conclusive high-level models of categorization and evaluation which generally apply to all creative domains in question.

One might organize approaches of computational creativity into three interrelated categories (Davies et al. 2015):tools that support human creativity (creativity support tools), programs that produce creative artifacts and approaches like "computer colleagues" that involve improvisational interaction with a user and can be seen in the context of co-creativity.

Creativity support tools "extend users capability to make discoveries or inventions from early stages of gathering information, hypothesis generation, and initial production, through the later stages of refinement, validation, and dissemination." (Schneiderman 2007, 22). According to three metaphors borrowed from Kumiyo Nakakoji (Nakakoji 2006) creativity support tools can 1. Improve already known creative abilities (they are "running shoes"). 2. Help a user to get knowledge about a specific domain and develop creative skills (they serve as "dumbbels"). 3. Enable an experience which is not possible without the tool ("skis"). Examples of creativity support tools are "iCanDraw" which provides a user with feedback and suggestions for drawing a human face from an existing image (Dixon, Prasad and Hammond 2010) or "MILA-S" which enables students to develop and evaluate conceptual models about ecological phenomena (Goel and Joyner 2015).[18] Creativity support tools in the musical domain may range from simple ear training programs (“running shoes“), up to sophisticated applications like the “Continuator” (Pachet 2003) or “Flow Machines” (Pachet, Roy and Ghedini 2013) which, by complex application of Markov models (Pachet and Roy 2011), aim to enhance individual creativity by man-machine interactions that attempt to imitate a user’s style (“skis”). The use of such extended formalisms, like Variable Markov Models (VMM), also led to systems of computer-aided improvisation (cf. Assayag and Dubnov 2004) which finally resulted in–to name a current prime example–OMax, which "creates a cooperation between several heterogeneous components specialized in real-time audio signal processing, high level music representations and formal knowledge structures." (Assayag 2016, 62).

These manifold approaches demonstrate that the abovementioned categories are to be seen more as fuzzy sets than easily separable areas like in the context of artifact producing programs or computer colleagues.

Nicholas Davis et al. rate computer colleagues as the “newest and perhaps most ambitious venture in the space of computational creativity” (Davis et al. 2015, 213) as they require complex methods for controlling the improvisational interaction with the user and the generation of creative contributions to the shared artifact. Exemplary approaches are the “Drawing Apprentice” (Davis et al., 2014) which collaborates with a user in abstract drawing on a shared canvas, and “Shimon” (Hoffman and Weinberg 2010), a marimba-playing robot which continuously adapts its improvisation and choreography while listening to a human co-performer.

Creative artifact producing programs besides the musical domain started with applications, like the forerunner of all chatbots, “ELIZA” (Weizenbaum 1966) and now cover a broad range of creative domains. A few recent applications include painting (Colton 2012), poetry generation (Misztal and Indurkhya 2014) story telling (Laclaustra et al. 2014), the generation of slogans (Tomašič, Žnidaršič and Papa 2014), generation of humorous puns (Valitutti, Stock and Strappavara 2009) and internet memes (Olivera, Costa and Pinto 2016), dance (Infantino et al. 2016) and even cooking (Shao, Murali and Sheopuri 2014). Creative computational approaches are also developed for the automated construction of conjectures and the proving of theorems in mathematics, cf. (Lenat 1977) and (Colton 2002).

Irrespective of their application field, computational creativity approaches can also be considered in regard to key concepts of creativity which were primarily developed by Margaret Boden.

Boden[19] distinguished between “psychological creativity” (P-creativity) which assesses creativity that is new with respect to its creator, whereas “historical creativity” (H-creativity) is recognized as new with respect to human history. Boden further distinguished between “exploratory creativity” that arises from a well-defined “conceptual space”, whereas “transformational creativity” happens when this conceptual space undergoes a radical transformation. Boden’s notion of the conceptual space may be seen in analogy to a search space in artificial intelligence, which defines the set of all possible solutions that satisfy a problem’s constraints. Graeme Ritchie (Ritchie 2006) points out that Boden did not define this term precisely and, among other things, suggests a hierarchical model in which the conceptual space consisting of “typical items” is a subset of “well-formed and logically possible items”, which is again a subset of all items which can be represented using the basic data type. Transformation happens here if the conceptual is extended out into the set of logical items, of course only if typical items are a proper subset of the logical items, meaning they are not identical. Ritchie illustrates this model by the game of chess: if “typical games” (i.e. the conceptual space) are a subset of "logically possible games," transformation will still happen in the context of valid chess moves; if the conceptual space is identical to the logically possible set of games, transformation can only happen if new moves are introduced which inevitably lead to a redesign of the game of chess.[20]

A widely applicable category by Boden is "combinatorial creativity" which generates novel concepts by combining familiar concepts in an unfamiliar way. A novel approach to computational creativity can be seen as analogous to "combinatorial creativity," i.e. "conceptual blending." Conceptual blending was developed as a theory of cognition by Gilles Fauconnier and Mark Turner (Fauconnier and Turner 1998; Fauconnier and Turner 2010) and has recently been adopted as a recognized branch of computational creativity.

Francisco C'mara Pereira offers the following definition:

"Blending is generally described as involving two input knowledge structures (the mental spaces) that, according to a given structure mapping, will generate a third one, called Blend. This new domain will maintain partial structure from the input domains and add emergent structure of its own." (Pereira 2007, 54)

One example in the musical domain is the system of Maximos Kaliakatsos-Papakostasa et al. (2016) in which the blend of two harmonic spaces (i.e. chord progressions) leads to the generation of novel harmonic spaces, which are adjusted according to the evaluation of a music expert, who interacts with the system through a Graphical User Interface (GUI).[21]

These concepts allow for a more detailed assessment of creativity in algorithmic composition and generative music systems if we now apply them to our previous subdivision of the simple relation of a subject (human) rating an object (computer) into the more complex relation of a user and/or society rating an artifact and/or the program that produces it.

Thus, from a composer's point of view the assessment of computational creativity can be seen in the context of P-creativity, whereas from the audience/society's point of view, the assessment would be on the level of H-creativity. Accordingly, a computer-generated artifact in the style of minimal music might be P-creative, although H-creativity might not be attributed to such an artifact generated after the rise of minimalism in the early 1970s.

Computational creativity provides many evaluation criteria which are often based on certain properties of the generated artifacts. Ritchie distinguishes among "typicality," "quality," and "novelty," which should be assessed by means of the following questions. Typicality: To what extent is the produced item an example of the artifact class in question? Quality: To what extent is the produced item a high-quality example of its genre? Novelty: To what extent is the produced item dissimilar to existing examples of its genre? (Ritchie 2007, 72). Simon Colton (Colton 2008) bases the assessment of a system not only on its artifacts but also on the underlying software (the program) which should, according to his "creative tripod," exhibit "skillful," "appreciative," and "imaginative" behavior. He also points to the fact that for the assessment of creativity, there are in most cases three parties involved, namely the computer, the programmer and the consumer, whereas "it is commonplace for people to attribute creativity to the programmer in addition [to] (or instead of) the software." (Colton 2008, 17). This notion is also thematized by Martin Mumford and Dan Ventura, who performed a case study to assess how people perceive and estimate the creativity of programs and generated artifacts (Mumford and Ventura 2015).

We can also locate programs and artifacts of computer music within the four levels of computational creativity defined by Diarmuid P. O’Donoghue, James Power, Sian O’Briain, Feng Dong, Aidan Mooney, Donny Hurley, Yalemisew Abgaz and Charles Markham (O’Donoghue et al. 2012, 152): “Direct Computational Creativity” (DCC), whereas ”…the outputs (artefacts or processes) display the novelty and quality attributes associated with creativity.”“Direct Self-Sustaining Creativity” (DSC), whereas “…the outputs are added to the inspiring set and serve to drive subsequent creative episodes.” “Indirect Computational Creativity” (ICC) that “…outputs a creative process—and that creative process is itself creative.” And at the most challenging level, “Recursively Sustainable Creativity” (RSC) “…where RCC learns from its own outputs to maintain its own creativity.”[22]

Considering computer music under these manifold criteria enables also to single out distinct factors of the creative process, as well as perceiving it as an emergent phenomenon arising from a complex interplay between, and often a deliberate boundary-crossing of, multifarious aspects of the interaction of composers and computers.

Footnotes

[1] Lisp can also be used for the control of sound synthesis, for example in the language Nyquist, see (Dannenberg 1997).

[2] For an essential overview of some of his–not only stochastic–approaches, see (Xenakis 1992).

[3] For a comprehensive study of his concepts, see his books (Barbaud 1971; Barbaud 1991).

[4] This includes style generation, herein defined as the modeling of a style musicologically understood as a proper genre in a historical or ethnological perspective.

[5] As expected, the knowledge-based approach was superior in solving the task, see (Phon-Amnuaisuk and Wiggins 1999).

[6] Mathematically, it is impossible to produce pure random numbers (cf. Chaitin 1975), thus one reverts again to deterministic algorithms with a complex system behavior such as cellular automata to simulate random properties.

[7] Often the term representation is in fact used in place of encoding: “An individual member of a population is represented as a string of symbols” (Biles 1995, 1), and vice versa: “The pitch encodings in ESAC resemble “solfege”: scale degree numbers are used to replace the movable syllables “do”, “re”, “mi”, etc.” (Bod 2002, 198).

[8] From the perspective of computer-aided musicology, the concept of "representation"–in a broader sense – can also be strongly linked to that of formalization, as outlined in (Acotto and Andreatta 2012).

[9] E.g. generating jazz chord progressions via Lempel-ZIV trees, cf. (Pachet 1999); another approach based on context-sensitivity is Kohonen's "Self Learning Musical Grammar", see (Kohonen 1987), (Kohonen 1989).

[10] Absolute pitch representation is more robust compared to an interval representation where numerical error propagates through all successive values.

[11] For other applications of neural networks in algorithmic composition, see (Todd 1989), (Bellgrad and Tsang 1994), (Schmidthuber 2002), (Sturm, Santos and Korshunova 2015).

[12] Cf. (Beyls 1989), (Beyls 1991), (Millen 1990), and (Hunt and Orton 1991)-

[13] Examples of multiple mapping can be found where visualisations of algorithms are subsequently translated into musical data, e.g. in the musical adaptation of the turtle graphics produced by Lindenmayer-systems. In this double translation, the aspect of self-similarity, essential to this algorithm class, is entirely lost (cf. Prusinkiewicz 1986).

[14] These are electronic art, computer art, digital art, computer-assisted/aided art, generative art, computer-generated generative art, evolutionary art, robot art, mechanical art, interactive art, computer interactive art, and virtual reality art.

[15] Design here is explicitly referring to the user interface and not the underlying architecture of the particular computer music languages. These dependencies are also reflected by software studies, a recent field dedicated to software design from a media and culture perspective, (cf. Fuller 2008).

[16] Interfaces encompass everything from conventional interfaces to concepts such as the monitoring and mapping of brainwaves, all in line with Varèse's vision: "I dream of instruments obedient to my thought…" (Varèse and Wen-chung 1966, 11). These interfaces represent more or less serious attempts to transcend the machine as intermediary and make it possible to shape musical structures directly from the power of imagination. For interfaces, see (Paradiso 1997); for "direct mind access", see Alvin Lucier's "Music for Solo Performer" from 1965 (cf. Straebel and Thoben 2014); newer approaches can be found, e.g. in (Miranda and Brouse 2005) or (Mullen 2011).

[17] Batch mode or batch processing denotes a mode of processing in which a set of instructions is sequentially executed without the possibility of interaction or manual intervention. This creates the following cycle: 1. Encoding the algorithm. 2. Entering the input data. 3. Executing the program. 4. Either accepting the result or restarting the process.

[18] "Conceptual models are abstract representations of the elements, relationships, and processes of a complex phenomenon or system." (Goel and Joyner 2015, 285).

[19] For the following categories of Boden, c.f. (Boden 2003) and (Boden 1999).

[20] For a formalized approach to these categories of Boden, see (Wiggins 2001) and (Wiggins 2003).

[21] For another category-theoretical framework of the creative process in the musical domain and a general discussion on conceptual blending, c.f. (Andreatta et al. 2013). For examples from other domains, see (Li et al. 2012, 11).

[22] Like in genetic programming (cf. Koza 2003).

References

Acotto, Edoardo, and Moreno Andreatta. "Between Mind and Mathematics. Different Kinds of Computational Representations of Music. " Mathématiques et sciences humaines, no. 199 (2012): 7-25.

Agon, Carlos, Olivier Delerue and Camilo Rueda. "Objects, Time and Constraints in OpenMusic." Proceedings of the ICMC 98 (1998).

Andreatta Moreno, Andrée Ehresmann, René Guitart, and Guerino Mazzola. "Towards a categorical theory of creativity." Proceedings of the Conference MCM 2013, Springer (2013): 19-37.

Assayag, Gérard, Michèle Castellengo and Claudy Malherbe. "Functional integration of complex instrumental sounds in music writing." Proceedings of the ICMC 85 (1985).

Assayag, Gérard, and Shlomo Dubnov, "Using Factor Oracles for machine Improvisation." Soft Computing–a Fusion of Foundations, Methodologies and Applications, 8, no. 9 (2004): 604-610.

Assayag, Gérard. "Improvising in Creative Symbolic Interaction." In Jordan B. L. Smith, Elaine Chew, and Gérard Assayag, Mathematical Conversations: Mathematics and Computation in Music Performance and Composition, World Scientific; Imperial College Press (2016): 61-74.

Backus, John. W., H. Stern, I. Ziller, R. A. Hughes, R. Nutt, R. J. Beeber, S. Best, R. Goldberg, L. M. Haibt, H. L. Herrick, R. A. Nelson, D. Sayre, and P. B. Sheridan. "The FORTRAN automatic coding system." Proceedings of the Western Joint Computer Conference (1957): 188-197.

Barbaud, Pierre. La Musique, Discipline Scientifique: Introduction Élémentaire à L'étude Des Structures Musicales. Paris: Dunod, 1971.

Barbaud, Pierre. Vademecum De L'ingénieur En Musique. Paris: Springer, 1993.

Barr, Avron, and Edward A. Feigenbaum. The Handbook of artificial intelligence. W. Kaufmann, 1981.

Bel, Bernard. "Migrating Musical Concepts: An Overview of the Bol Processor." Computer Music Journal 22, no. 4 (1998): 56-64.

Bellgard, Matthew I., and C. P. Tsang. "Harmonizing Music the Boltzmann Way." Connection Science 6, no. 2-3 (1994): 281-97.

Beyls, Peter. "The musical universe of cellular automata." Proceedings of the 1989 International Computer Music Conference (1989): 34-41.

Beyls, Peter. "Self-organising control structures using multiple cellular automata." Proceedings of the 1991 International Computer Music Conference (1991): 254-57.

Biles, John A. " GenJam: a genetic algorithm for generating jazz solos." Proceedings of the 1994 International Computer Music Conference (1994): 131-37.

Biles, John A. "GenJam Populi: training an IGA via audience-mediated performance." Proceedings of the 1995 International Computer Music Conference (1995): 347-48.

Bod, Rens. " A Unified Model of Structural Organization in Language and Music." Journal of Artificial Intelligence Research 17 (2002): 289-308.

Boden, Margaret A. "Creativity and artificial intelligence." Artificial Intelligence 103, no. 1-2 (1998): 347-56.

Boden, Margaret A. “Computational models of creativity.‟ In Sternberg , Robert J. ed. Handbook of Creativity. 351-373. Cambridge: Cambridge University Press,1999.

Boden, Margaret A. The creative mind: myths and mechanisms. London ; New York, NY: Routledge, 2003.

Boden, Margaret. A., Edmonds, Ernest. “What is Generative Art?” Digital Creativity, 20 (1-2). (2010): 21-46.

Boring Edwin G. "Intelligence as the Tests Test it." New Republic 3 (1923): 35-37.

Brooks, Frederick P., A. L. Hopkins, P. G. Neumann, and W. V. Wright. "An experiment in musical composition." IRE Transactions on Electronic Computers EC-6, no. 3 (1957)

Cage, John. Silence: Lectures and writings, 50th anniversary edition. Middletown: CT Wesleyan University Press, 2013.

Chai, Wei, and Barry Vercoe. "Folk music classification using hidden Markov models." Proceedings International Conference on Artificial Intelligence (2001).

Chaitin, Gregory J. "Randomness and Mathematical Proof." Scientific American 232, no. 5 (1975): 47-52.

Chemillier, M. "Toward a formal study of jazz chord sequences generated by Steedman’s grammar." Soft Computing 8, no. 9 (2004).

Colton, Simon. Automated theory formation in pure mathematics. London: Springer, 2002.

Colton, Simon. "Creativity versus the perception of creativity in computational systems." AAAI

Spring Symposium: Creative Intelligent Systems (2008): 14-20.

Colton, Simon. "The Painting Fool: Stories from Building an Automated Painter." Computers and Creativity (2012): 3-38.

Cope, David. "An Expert System for Computer-Assisted Composition." Computer Music Journal 11, no. 4 (1987): 30-46.

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

Cope, David. The Algorithmic Composer. Madison, WI: A-R Ed., 2000.

Cope, David. Virtual Music. Cambridge, MA: MIT Press, 2001.

Cope, David. Experiments in Musical Intelligence. Middleton, WI: A-R Editions, 2014.

Csikszentmihalyi, Mihali. "Implications of a Systems Perspective for the Study of Creativity." In Sternberg, Robert J. Handbook of Creativity. Cambridge: Cambridge University Press, 1999, 313-336.

Dannenberg, Roger B. "Machine Tongues XIX: Nyquist, a Language for Composition and Sound Synthesis." Computer Music Journal 21, no. 3 (1997): 50-60.

Davis, Nicholas, Yanna Popova, Ivan Sysoev, Chih-Pin Hsiao, Dingtian Zhang, and Brian Magerko. "Building Artistic Computer Colleagues with an Enactive Model of Creativity‟. Proceedings of the Fifth International Conference on Computational Creativity (2014): 38-45.

Davis, Nicholas, Chih-Pin Hsiao, Yanna Popova, and Brian Magerko. "An Enactive Model of Creativity for Computational Collaboration and Co-creation." Creativity in the Digital Age, Springer Series on Cultural Computing(2015): 109-33.

Dixon, Daniel, Manoj Prasad, and Tracy Hammond. "iCanDraw: using sketch recognition and corrective feedback to assist a user in drawing human faces.‟ Proceedings of the SIGCHI conference on human factors in computing systems(2010): 897-906.

Dorin, Alan. "Boolean Networks for the generation of rhythmic structure." Proceedings of the 2000 Australian Computer Music Conference (2000): 38-45.

Fauconnier, Gilles, and Mark Turner. "Conceptual Integration Networks." Cognitive Science 22, no. 2 (1998): 133-87.

Fauconnier, Gilles, and Mark Turner. The way we think: conceptual blending and the mind’s hidden complexities. New York: Basic Books, 2010.

Fuller, Matthew. Software Studies: A Lexicon. Cambridge (Massachusetts): MIT, 2008.

Galanter, Philip. "Philip Galanter Philip Galanter What is generative art? Complexity theory as a context for art theory." GA2003–6th Generative Art Conference (2003).

Galanter, Philip. "Generative Art after Computers." GA2012–15th Generative Art Conference (2012): 271-82.

Gardner, Howard. Frames of mind The theory of multiple intelligences. New York: Basic Books, 2011.

Gartland-Jones, Andrew, and Peter Copley. "The Suitability of Genetic Algorithms for Musical Composition." Contemporary Music Review 22, no. 3 (2003): 43-55.

Goel, Ashok K., and David A. Joyner. "Impact of a Creativity Support Tool on Student Learning about Scientific Discovery Processes." Proceedings of the Sixth International Conference on Computational Creativity (2015): 284-91.

Goertzel, Ben. The Hidden Pattern: a Patternist Philosophy of Mind. Boca Raton: BrownWalker Press, 2006.

Goßmann, Joachim. "Towards an auditory Representation of Complexity." Proceedings of ICAD 05-Eleventh Meeting of the International Conference on Auditory Display (2005): 264-68.

Hoffman, Guy, and Gil Weinberg. "Gesture-based human-robot Jazz improvisation." 2010 IEEE International Conference on Robotics and Automation (2010): 582-587.

Horner, Andrew, and David E. Goldberg. "Genetic Algorithms and Computer-Assisted Music Composition." Proceedings of the 1991 International Computer Music Conference (1991): 479-482.

Hughes, David W. (1991). "Grammars of Non-Western Musics: A Selective Survey." In Representing Musical Structure, edited by Howell, Peter, Robert West, and Ian Cross, 327-362. London: Academic Press, 1991.

Hunt, Andy, R. Kirk, and R. Orton. "Musical Applications of a Cellular Automata Workstation." Proceedings of the 1991 International Computer Music Conference (1991): 165-68.

I. Infantino, A. Augello, A. Manfré, G. Pilato, and F. Vella. "ROBODANZA: Live Performances of a Creative Dancing Humanoid.“ Proceedings of the Seventh International Conference on Computational Creativity (2016): 388-395.

Kaliakatsos-Papakostasa, Maximos, Roberto Confalonieri, Joseph Cornelic, Asterios Zacharakisa, and Emilios Cambouropoulosa. "An Argument-based Creative Assistant for Harmonic Blending." Proceedings of the Seventh International Conference on Computational Creativity (2016): 330-37.

Kippen, Jim, and Bernard Bel. "The identification and modelling of a percussion ‛language, ‛ and the Emergence of Musical Concepts in a machine-learning experimental set-up." Computers and the Humanities 23, no. 3 (1989): 199-214.

Kohonen, Teuvo. "Self-learning inference rules by dynamically expanding context." Proceedings of the IEEE First Annual International Conference on Neural Networks (1987).

Kohonen. "A Self-learning Musical Grammar, or ʽassociative Memory of the Second Kind’" International Joint Conference on Neural Networks, 1989.

Koza, John R. Genetic programming. Cambridge, Mass.: MIT Press, 2003.

Iván M. Laclaustra, José L. Ledesma, Gonzalo Méndez, and Pablo Gervás. "Kill the Dragon and Rescue the Princess: Designing a Plan-based Multi-agent Story Generator.‟ Proceedings of the Fifth International Conference on Computational Creativity (2014): 347-350.

Laske, Otto E. "In Search of a Generative Grammar for Music." Perspectives of New Music 12, no. 1/2 (1973): 351-378.

Laurson, Mikael, "PatchWork: A Visual Programming Language and Some Musical Applications". PhD diss., Helsinki: Sibelius Academy, 1996.

Lenat, John B. "Automated theory formation in mathematics." IJCAI’77 Proceedings of the 5th international joint conference on Artificial intelligence 2 (1977).

Lerdahl, Fred, and Ray Jackendoff. A Generative Theory of Tonal Music. Cambridge, Mass.: MIT Press, 2010.

LeWitt, Sol. "Paragraphs on Conceptual Art." Artforum, June 1967.

Li, Boyang, Alexander Zook, Nicholas Davis, and Mark O. Riedl. "Goal-Driven Conceptual Blending: A Computational Approach for Creativity‟, Proceedings of the Third International Conference on Computational Creativity (2012): 9-16.

McCartney, James. "SuperCollider: A new real time synthesis language." Proceedings of the 1996 International Computer Music Conference (1996): 257-58.

Millen, Dale. "Cellular Automata Music." Proceedings of the 1990 International Computer Music Conference (1990): 314-16.

Miranda, Eduardo Reck, and Andrew Brouse. "Interfacing the Brain Directly with Musical Systems: On Developing Systems for Making Music with Brain Signals." Leonardo 38, no. 4 (2005): 331-36.

Misztal, Joanna, and Bipin Indurkhya. "Poetry generation system with an emotional personality‟. Proceedings of the Fifth International Conference on Computational Creativity (2014): 72-81.

Moray, Allen. "Harmonising chorales in the style of Johann Sebastian Bach." master’s thesis, University of Edinburgh, 2002.

Moray, Allen, and Christopher K. I. Williams. "Harmonising Chorales by Probabilistic Inference." Advances in Neural Information Processing Systems 17 (2005): 25-32.

Mozer, Michael C. "Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing." Connection Science 6, no. 2-3 (1994): 247-80.

Mullen, Tim, Richard Warp, and Adam Jansch. "Minding the (Transatlantic) Gap: An Internet-Enabled Acoustic Brain-Computer Music Interface." Proceedings of the International Conference on New Interfaces for Musical Expression(2011): 469-72.

Mumford, Martin, and Dan Ventura. "The man behind the curtain: Overcoming skepticism about creative computing.‟ Proceedings of the Sixth International Conference on Computational Creativity (2015): 1-7.

Nakakoji, Kumiyo. "Meanings of tools, support, and uses for creative design processes." International design research symposium (2006): 156-65.

Neumann, John von. First draft of a report on the EDVAC. Philadelphia, PA: Moore School of Electrical Engineering, University of Pennsylvania, 1945.

Nierhaus, Gerhard. Algorithmic Composition: Paradigms of Automated Music Generation. Wien: Springer, 2009.

Nierhaus, Gerhard. Patterns of Intuition: Musical Creativity in the Light of Algorithmic Composition. Dordrecht: Springer, 2015.

O’Donoghue, Diarmuid P., James Power, Sian O’Briain, Feng Dong, Aidan Mooney, Donny Hurley, Yalemisew Abgaz, and Charles Markham. ‟Can a Computationally Creative System Create Itself? Creative Artefacts and Creative Processes‟ Proceedings of the Fifth International Conference on Computational Creativity (2012): 146-154.

Oliveira, Hugo Gonçalo, Diogo Costa, and Alexandre Miguel Pinto. "One does not simply produce funny memes! – Explorations on the Automatic Generation of Internet humor.‟ Proceedings of the Seventh International Conference on Computational Creativity (2016): 238-245.

Pachet, François. "Surprising Harmonies." International Journal on Computing. Anticipatory Systems 4 (1999).

Pachet, François. "The Continuator: Musical Interaction With Style." Journal of New Music Research 32, no. 3 (2003): 333-41.

Pachet, François, and Pierre Roy. "Markov constraints: steerable generation of Markov sequences." Constraints 16, no. 2 (2010): 148-72.

Pachet, François , Pierre Roy, and Fiammetta Ghedini. "Creativity through Style Manipulation: the Flow Machines project." Marconi Institute for Creativity Conference (MIC 2013) 80, 2013.

Papadopoulos, George, and Geraint Wiggins. "A Genetic Algorithm for the Generation of Jazz Melodies." Proceedings of STEP 98 (1998): 7-9.

Paradiso, Joseph A. "Electronic Music: New Ways to Play." IEEE Spectrum 34, no. 12 (1997): 18-30.

Paynter, John. Companion to Contemporary Musical Thought. London: Routledge, 1992.

Pearson, Matt. Generative Art: A Practical Guide Using Processing. Manning Publications, 2011.

Pereira, Francisco C'mara. Creativity and artificial intelligence: a conceptual blending approach. Berlin: Mouton de Gruyter, 2007.

Phon-Amnuaisuk, Somnuk, and Geraint Wiggins. "The Four-Part Harmonisation Problem: A Comparison between Genetic Algorithms and a Rule-Based System." Proceedings of the AISB’99 Symposium on Musical Creativity (1999): 28-34.

Pinkerton, Richard C. "Information Theory and Melody." Scientific American 194, no. 2 (1956): 77-87.

Pope, Rob. Creativity: Theory, History, Practice. London, New York: Routledge, 2005.

Prusinkiewicz, Przemyslaw. "Score generation with L-systems." Proceedings of the 1986 International Computer Music Conference (1986): 455-57.

Ritchie, Graeme. "The transformational creativity hypothesis." New Generation Computing 24, no. 3 (2006): 241-266.

Ritchie, Graeme. "Some Empirical Criteria for Attributing Creativity to a Computer Program." Minds and Machines 17, no. 1 (2007): 67-99.

Roads, Curtis. The Computer Music Tutorial. MIT Press, 1996.

Roads, Curtis. Composing Electronic Music: A New Aesthetic. Oxford: Oxford University Press, 2015.

Roads, Curtis, and Paul Wieneke. "Grammars as Representations for Music." Computer Music Journal 3, no. 1 (1979): 48-55.

Rodet, Xavier, and Pierre Cointe. "FORMES: Composition and Scheduling of Processes." Computer Music Journal 8, no. 3 (1984): 32-50.

Rohrmeier, Martin. "Towards a generative syntax of tonal harmony." Journal of Mathematics and Music 5, no. 1 (2011): 35-53.

Rojas, Raúl, Cüneyt Göktekin, Gerald Friedland, Mike Krüger, Ludmila Scharf, Denis Kuniß, and Olaf Langmack. "Konrad Zuses Plankalkül — Seine Genese und eine moderne Implementierung." In Hellige, Hans Dieter, ed. Geschichten der Informatik Visionen, Paradigmen, Leitmotive. 215-35. Berlin, Heidelberg: Springer Berlin, 2004.

Eck, Douglas, and Jürgen Schmidhuber. "Learning the Long-Term Structure of the Blues." Artificial Neural Networks — ICANN 2002 Lecture Notes in Computer Science (2002): 284-89.

Schulze, Walter, Brink van der Merwe. "Music Generation with Markov Models." IEEE Multimedia 18, no. 3 (2011): 78-85.

Shane, Legg, and Marcus Hutter. "A Collection of Definitions of Intelligence." Proceedings of the 2007 conference on Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms: Proceedings of the AGI Workshop 2006 (2006): 17-24.

Shao, Nan, Pavankumar Murali, and Anshul Sheopuri. "New Developments in Culinary Computational Creativity.‟Proceedings of the Fifth International Conference on Computational Creativity (2014): 324-327.

Shepard, Roger N. "Geometrical approximations to the structure of musical pitch." Psychological Review 89, no. 4 (1982): 305-33.

Shneiderman, Ben. "Creativity support tools: accelerating discovery and innovation." Communications of the ACM 50, no. 12 (2007): 20-32.

Steedman, Mark J. "A Generative Grammar for Jazz Chord Sequences." Music Perception: An Interdisciplinary Journal 2, no. 1 (1984): 52-77.

Sternberg, Robert J. Handbook of Creativity. Cambridge: Cambridge University Press, 1999.

Sternberg, Robert J., and William Salter. "Conceptions of Intelligence." In Sternberg, Robert J., ed. Handbook of human intelligence. Cambridge: Cambridge University Press, 1982.

Steyn, Jacques. Structuring music through markup language: designs and architectures. Hershey, PA: Information Science Reference, 2013.

Stockhausen, Karlheinz. " ...wie die Zeit vergeht...." In Stockhausen, Karlheinz. Texte zur elektronischen und instrumentalen Musik. Edited by Dieter Schnebel. Vol. 1. Köln: Dumont Schauberg, 1963.

Stockhausen, Karlheinz. "Four Criteria of Electronic Music." In Stockhausen, Karlheinz, and Robin Maconie. Stockhausen on Music. London: Marion Boyars, 1989.

Straebel, Volker, and Wilm Thoben. "Alvin Lucier's Music for Solo Performer: Experimental music beyond sonification." Organised Sound 19, no. 01 (2014): 17-29.

Stravinsky, Igor. Igor Stravinsky, An autobiography. New York: Norton Library, 1962.

Sturm, Bob, João Felipe Santos, and Iryna Korshunova. 2015. "Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units." 16th International Society for Music Information Retrieval Conference, late-breaking demo session (2015).

Taube, Heinrich. "Common Music: A Music Composition Language in Common Lisp and

CLOS." Computer Music Journal 15, no. 2 (1991): 21-32.

Taube, Heinrich. Notes from the Metalevel: Introduction to Algorithmic Music Composition. London: Taylor & Francis Group, 2004.

Todd, Peter M. "A Connectionist Approach to Algorithmic Composition." Computer Music Journal 13, no. 4 (1989): 27-43.

Polona Tomašič, Martin Žnidaršič, and Gregor Papa. "Implementation of a Slogan Generator.‟ Proceedings of the Fifth International Conference on Computational Creativity (2014): 340-343.

Turing, Allan M. "On Computable Numbers, with an Application to the Entscheidungsproblem." Proceedings of the London Mathematical Society, 2nd ser., 42 (1937): 230-65.

Vaggione, Horacio. "Some Ontological Remarks about Music Composition Processes." Computer Music Journal 25, no. 1 (2001): 54-61.

Valitutti, Alessandro, Oliviero Stock, and Carlo Strapparava. "GraphLaugh: A tool for the interactive generation of humorous puns." 2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops(2009): 1-2.

Weizenbaum, Joseph. "ELIZA---a computer program for the study of natural language communication between man and machine." Communications of the ACM 9, no. 1 (1966): 36-45.

Varèse, Edgard and Chou Wen-chung. "The Liberation of Sound," Perspectives of New Music 5, no. 1 (1966): 11-19.

Ward, Adrian, and Geoff Cox. "How I Drew One of My Pictures: * or, The Authorship of Generative Art." GA1999–2th Generative Art Conference (1999).

Watts, Alan. Nature, Man and Woman. New York: Vintage Books, 1970.

Wechsler, David. Measurement and Appraisal of Adult Intelligence. 4th ed. Baltimore: Williams & Wilkins, 1958.

Wiggins, Geraint. "Towards a more precise characterisation of creativity‟ in AI.‟ In R. Weber, C. G. von Wangenheim, eds, Case-based reasoning: Papers from the workshop programme at ICCBR 01. Vancouver: Navy Center for Applied Research in Artificial Intelligence (2001).

Wiggins, Geraint. "Categorising creative systems." Proceedings of Third (IJCAI) Workshop on Creative Systems: Approaches to Creativity in Artificial Intelligence and Cognitive Science (2003).

Wright, Matthew. "Open Sound Control: an enabling technology for musical networking." Organised Sound 10, no. 03 (2005): 193-200.

Xenakis, Iannis. Formalized Music: Thought and Mathematics in Composition. Rev. ed.

Harmonologia Series, no. 6. Stuyvesant, NY: Pendragon Press, 1992.

zoogle-video#handleVimeoPostMessage">

- Log out

Textise: Back to top