DRADEK

Studies in Philosophy of Literature, Aesthetics,
and New Media Theories

Vol. VIII Num. 1 2022

ISSN 2465-1060
[online]

Creativity in the Light of AI

Edited by
Fabio Fossa, Caterina Moruzzi, Mario Verdicchio

powered by

1343

UNIVERSITÀ DI PISA

Comitato Direttivo/Editorial Board:
Danilo Manca (Università di Pisa, editor in chief), Francesco Rossi (Università di Pisa), Alberto L. Siani (Università di Pisa).

Comitato Scientifico/Scientific Board
Leonardo Amoroso (Università di Pisa), Christian Benne (University of Copenhagen), Andrew Benjamin (Monash University, Melbourne), Fabio Camilletti (Warwick University), Luca Crescenzi (Università di Trento), Paul Crowther (NUI Galway), William Marx (Université Paris Ouest Nanterre), Alexander Nehamas (Princeton University), Antonio Prete (Università di Siena), David Roochnik (Boston University), Antonietta Sanna (Università di Pisa), Claus Zittel (Stuttgart Universität).

Comitato di redazione/Executive Committee:
Alessandra Aloisi (Oxford University), Daniele De Santis (Charles University of Prague), Agnese Di Riccio (The New School for Social Research, New York), Fabio Fossa (Università di Pisa), Beatrice Occhini (Università di Napoli "L'Orientale"), Elena Romagnoli (Scuola Normale Superiore di Pisa), Marta Vero (Università di Pisa, journal manager).

ODRADEK. Studies in Philosophy of Literature, Aesthetics, and New Media Theories.
ISSN 2465-1060 [online]

Edited by Università di Pisa

License Creative Commons
Odradek. Studies in Philosophy of Literature, Aesthetics and New Media Theories is licensed under a Creative Commons attribution, non-commercial 4.0 International.

Further authorization out of this license terms may be available at http://zetesisproject.com or writing to: zetesis@unipi.it.

Layout editor: Marta Vero
Volume Editor: Alberto Frigo

ODRADEK

Studies in Philosophy of Literature, Aesthetics, and New Media Theories

Vol. VIII Num. 1 2022

ISSN 2465-1060
[online]

Creativity in the Light of AI

Edited by
Fabio Fossa, Caterina Moruzzi, Mario Verdicchio

powered by

Università di Pisa

Computation, Creativity, and Improvised Music

René Mogensen

Abstract

I investigate the intersection of the concepts ‘creativity’ and ‘computation’ in the context of improvised music. While these concepts are commonly thought of as opposites, I argue that they can be intimately interlinked when humans and computational systems contribute to improvised music performance. I take human creativity and computational creativity to be categorically different. However, computational creativity in improvised music may be grounded in a ‘knowing how’ to improvise computationally and may contribute to the *distributed creativity* of a human-machine performance system. The semantics of humans and computational systems are of different categories and their respective musical ‘purposefulness’ are also categorically different. However, these differences allow interaction; and when engaged in group improvisation both humans and computational systems can be engaged in contributing to a *co-creative* improvised music performance.

1. Creativity and computation

How can the ideas of 'creativity' and 'computation' intersect? In common language-use these two terms frequently seem to be used as though they are in opposition: 'creativity' is often used in relation to the arts and is by implication related to activity which is fuelled by human fantasy, 'inspiration', or imagination. In this sense 'creativity' is considered to be evident in the invention of new artefacts or concepts with aesthetic impact. On the other hand, 'computation' is often associated with mathematical procedures which are applied to give repeatable solutions to practical tasks and problems in a positivistic sense. In contrast to the idea of opposition, I argue that creativity and computation can sometimes be interlinked and intimately interdependent.[1]

Carnovalini and Rodà point out that the expectations of deterministic behaviour that we have of computers "seems to be the exact opposite of our understanding of the concept of creativity".[2] Of course it is still an open question as to how much of human behaviour, and also human creativity, is deterministic; but I will not attempt to address that question here. For present purposes it is sufficient to point out that the unknown degree of determinism in human creativity means that determinism does not necessarily exclude the possibility of creativity

---

[1] By using the prefix *inter-* in the terms 'interlinked' and 'interdependent' I want to emphasise that the links and dependencies can apply both from computation to creativity as well as from creativity to computation.

[2] Carnovalini and Rodà (2020), p. 2.

in computational systems.

As an example of a common misperception of mathematical computation being in opposition to ‘creativity’, we could consider that many primary school children might not perceive mathematics as a ‘creative’ subject when, for example, they have to memorise fundamental operations such as those expressed in multiplication tables. This perception may of course vary according to the teaching methods used. In any case, not every application of a computational operation necessarily yields a creative result. This is similar to the sense that playing a note on an instrument is not necessarily a creative result. However, playing an original sequence of notes may be perceived as creative. Similarly, the application of fundamental mathematical operations into larger complexes of operations and interpreting these complexes, as well as their applications in theories and practices, may be considered creative. My impression is that most mathematicians would probably agree that their field of work requires creativity, and yet computation is a significant component of that field. From their perspective then, creativity and computation are interlinked. Composed music works are often considered creative and some composers, who practice ‘integral serialism’ have used mathematical relationships as tools for music composition, and so also navigate an interlinking and interdependence of computation and creativity (more on this in Section 4).

The ‘interlinking’ and ‘interdependence’ examined here is from a broad perspective that crosses performative musical creativity and computer systems. Examples of the interdependence of creativity and computation can occur in improvised music performances by groups consisting of musicians and computer-based improviser systems that are performing together. Performances by such groups are not necessarily deemed to be 'creative' by interested listeners. But I take human music improvisation as being at least potentially 'creative'. My present discussion uses the premise that improvisation can be creative in principle, rather than it being necessarily so in specific cases; therefore, I limit the present scope to improvised performances that are considered 'creative' by interested listeners and ignore cases of improvised performances which might be considered pedestrian or otherwise non-creative by audiences and/or participating performers.

The interlinking of creativity and computation leads to the possibility of human-computer *co-creative* results where there may be partnerships between humans and computational systems. Distributed creativity, including both musicians and computer systems, may 'emerge' in such partnerships: the computational system may contribute to group improvisation where "improvisation can be productively understood as 'listening-while-performing' – a clear-cut example of the pervasive ecological principle of perception-action coupling in which playing informs listening, and listening informs playing".[3]

---

[3] Linson and Clarke (2017), p. 63.

Perhaps the concept of 'listening' does not apply to a computational system, since the term tends to imply some kind of conscious interpretation of sound. But in practice, computational 'listening' is often reduced to the calculation of features ('feature extraction') of microphone input(s), where those features are used to represent the music that is interpreted by the system.[4] So, computational listening is not the same as human listening, but computational systems can 'interpret' sound without necessarily having consciousness. 'Emergence' is also a controversial concept, but in this context I use it to denote a system-effect of the actions of the self-organising members (human and/or computational) of the improvising music group. In this usage the 'system' is the performing group in the performance context which includes the venue and audiences, and which may in turn be influenced by the larger intertextual network.[5]

2. Categories of human and computational creativity

Human improvisation may be closely tied to cognition. Dror and Harnad argue that machines "can sometimes contribute to human cognition, but that does not make them cognizers" because

---

[4] Examples of implementations employing 'feature extraction' include many of the systems surveyed in Gifford *et al.* (2018) as well as the system documented in Mogensen (2020).

[5] I use the idea of 'intertextual network' in the sense of Klein (2005).

machines do not have 'mental states'; they seem to equate 'cognition' with 'mental states', and so in their view machines may extend human cognition but are not inherently cognitive systems.[6] This leads to an interpretation of distributed cognition and consequently also distributed creativity as an environmental extension of human cognition and creativity.

Computational creativity may be dependent on human cognitive involvement. However, even without 'mental states', computational systems can form a category of creativity as *contributors* to distributed creativity results. It may be that human creativity is also dependent on functional involvement as a contributor to distributed creativity. For example, Boden's category of Historical Creativity seems to imply that her 'transformational creativity' only occurs in a social and ecological context where the 'creativity' of the individual person is only possible as part of a larger context and on a timescale of human history.[7] Boden's Personal Creativity seems more akin to learning, perhaps experiential learning, since the 'newness' of Boden's personally creative act is a personal discovery.

In other words, my claim is that computational creativity can belong to a more general class of *contributors* to ecological, or distributed, creativity that also includes human creativity. I take human creativity and computational creativity to be categorically different. Computational creativity, as

---

[6] Dror and Harnad (2008), p. 1.

[7] Boden (2004).

a contributor to group creativity, can be functionally and operationally distinct from human creativity. Computational creativity does not necessarily model human creativity, even if it is engaged in contributing to an activity such as music-making which is often considered a quintessentially human activity. In what follows here, I examine this claim in the context of group music improvisation, where such a group performance can be a forum for human-computer co-creativity.

3. Computation and (human) cognition

Given that 'computation' and 'creativity' are two distinct concepts, although not opposites nor necessarily conflicting, then how can the referents of these concepts interact in the music performance situation? To examine this we might first ask: what does it mean for something to be computational? I presently take the view that a 'computer' is something that executes an encoding of a 'computation'. This gives an important distinction: the computer has a time dimension; but the computation itself is an abstract fact (or formally: a theorem) of a particular consistent formal system which in turn may not be time-dependent.

Computation in the mathematical sense may in principle be a static 'truth', since a result of any particular computation is not dependent on when, nor by whom, it is computed historically; rather it is usually proposed as a logical (or constant) truth, within a given formal system. The validity of a formal proof is usually considered to be a-historical even if it takes mathematicians many years (or even generations) to find that proof. Consequently, 'computations' are accepted as perpetually valid (also retroactively) if they are sound and coherent, again regardless of the historical position of their formulation. This can be the case even when computational results cannot be predicted without performing the computational process.[8]

We may allow these terms to designate concept families, in Wittgenstein's sense, so that the terms cover families of computational system-types and computation-types. A 'computer' can then be understood as an implementation (or realisation) of a member of the family of computational system-types which executes an encoding of a member of the family of computation types. So, while the mathematical 'truth' of a particular computation may be considered universally valid independently of a time-dimension, the computer which realises the computation in some encoded form, exists and functions during a particular time-frame.

Given this understanding of terms, humans can be 'computers', in the sense of an older (pre-digital computer) usage of the term, in that they can execute an encoded computation. The encoding of a computation may alternatively be placed in a machine,

---

[8] In other words, the 'stopping problem': there is no Universal Turing Machine which can predict whether another arbitrary Turing Machine will stop, as shown in Turing's 1936 paper (see also Petzold, 2008). Current digital computers are examples of Universal Turing Machines.

such as a digital computer; but so far this can only be done by programming the machine. While a new mathematical proof may be produced by a computer, the encoding necessary to arrive at that proof (i.e. a kind of meta-proof) must be encoded by a human. So, the instructions (the encoded computation) in a Universal Turing Machine have to be generated from some meta-machine which in this case would be one or more humans.

Is such a meta-machine necessarily human? Computer code (a computer program) which generates other computer code is already a reality, but at what point the earlier code qualifies as an 'originating' meta-machine is an open question. Perhaps identifying an originating meta-machine is a 'chicken-or-egg first' question, but the theory of evolution tells us that homo sapiens have evolved from earlier species. If machines evolve would this mean that their evolution would include some 'originating' meta-machine, a kind of '(missing) link' between biology and machine? I would not consider a Turing Machine to be a species in the biological sense, but identifying an originating computational meta-machine is an open question; and this meta-machine would by definition be a computational creativity since the criteria for identifying it would be its creative output.

Bringsjord, Bello, and Ferrucci[9] argued for a 'Lovelace Test' which a computational creativity could only pass if it could 'originate things'. In effect the 'Lovelace Test' demands that the computational

---

[9] See Bringsjord - Bello - Ferrucci (2000).

system gives output that cannot be explained by analysis of its formal system. In other words, the computational creativity system must give some kind of emergent output that it cannot generate within its formal definition. This seems to be a paradoxical demand, and so they concluded that it is unlikely, if not impossible, that a Universal Turing Machine could be an 'originating' machine.

However, computation as such may well be a characteristic that is specific to human cognitive interaction. If so, computation can be understood as a method by which we structure at least some cognitive activity. That is to say that 'computation' is a characteristic approach to human perception and interaction with the world; but it is a characteristic that is not necessarily shared by non-human entities or the rest of the world. If there is an interdependence between computation and human cognition, then perhaps something 'computational' only occurs when we apply the computational method in our cognitive involvement with a delimited system. Another way to say this is that human interpretation is necessary for 'computation'.

4. Computation and music

If we accept that human interpretation is necessary for computation then for computation to occur in music there must be an associated activity which includes human cognitive involvement.

When we interpret 'computational output', we align our thinking with non-living 'computers' and through such alignment we use these 'computers' to execute processes that we interpret as 'problem solving processes'. This does not deny the existence of physical processes which we can consider computational, but what I am suggesting is that these processes only give outputs that are computational results when humans are involved in interpreting these outputs as 'computational'. For example: in a group music improvisation performance, where computational agents may contribute to co-creative results with human performers,[10] the interpretation by humans occurs through the performers (and audience) listening to the computationally generated music. Responses to the computational outputs occur in the performers' musical actions, as well as in the emotive and/or analytical interpretations by audiences.

Perhaps I should make it clear that I am using a limited idea of 'problem solving processes' in the previous paragraph. I am not attempting to imply that music is a 'problem' to be solved nor that music-making is necessarily a problem-solving process. But a formal system (such as a Universal Turing Machine) goes through a process over time that gives computed results, where such results are solving the *formal* 'problems' or kinds of 'problems' that the formal system is constructed to address. In a music-generating system, the computer (a formal system)

---

[10] A number of examples of such computational agents are surveyed in the paper by Gifford *et al.* (2018) which I return to in Section 6.

is expected to resolve the transformation of input and generation of output over time according to its formal structure and thereby 'solving' its formal 'problems'. While such solutions to formal problems may result in sound, the perception of the musical qualities of that sound by listeners is not necessarily a problem-solving exercise in the formal sense.

Let us consider two prototype categories: 1. humanly-motivated computation; and 2. non-humanly motivated computation. By 'motivated' I mean *intrinsically* motivated so that action is taken that is at least partly driven by imperatives which are internal to the agent taking that action. In both prototype categories a 'computation' is a delineated system which encodes sufficient processes to give outputs which may represent a possible solution (or set of solutions) to a specific defined problem. These two prototype categories can be related to two other categories which we can consider with regards to a broad concept of sound (including music and sonic arts): 1. sound that is organised by human intentions (or motivations); and 2. sound that is organised, or occurs, without human intentions.

In the case of sound organisation, a computational system can take the form of an algorithm on paper, as was used for example in the 1950s 'integral serialism' where the composers 'computed' their music on paper. I use 'integral serialism' in the sense used by Brindle,[11] who quotes composer Luigi Nono referring to "so-called 'totally organized' methods of

---

[11] Brindle, (1987), Chapter 5 and (1967), Chapter 15.

composition".[12] The expression 'totally organized' in this case implies computability. The goal of integral serialism was arguably to create codified generative systems for music that were potentially interactive with, and appealing to human musical sensibilities.

But these composers' abilities and efforts to effect computational solutions to compositional problems did not define those same composers as purely 'computational' beings; it seems instead that they thought of the action of computing as a necessary (at that time) but insufficient part of being a human music creator (or composer). It was the involvement of the composers with the computational systems that made the music of 'integral serialism' possible. In cases where digital computers are active in a process, the understanding that these computers solve problems is dependent on the involvement of humans — computation only occurs when humans are using these computers and paying attention to (perhaps interpreting) the outputs of the computers at some point in time.

An example to illustrate the claim that interpretation may be crucial for something to be 'computational' is as follows: we see the Sun as an essential energy source for life on Earth. We could also attempt to interpret the Sun as an enormous computer, that is programmed to solve the problems of nuclear fission and fusion in order to change the atoms in its mass over the course of billions of years. These two interpretations express two different forms of cognitive involvement with the Sun. The second

---

[12] Brindle (1987), p. 21.

interpretation categorises the Sun as 'computational', the first does not. The computational interpretation sees the Sun as an example of a computation in time as formulated by a computational system which may be part of the science of Physics. A physical process is interpreted as being 'computational' through human engagement. This example of interpreting the Sun expresses a post-positivist view which does not seem controversial. An interpretation of the Sun as a computational system, in our sense, is clearly dependent on our cognitive involvement.

5. Group music improvisation

If we take music improvisation as a (non-verbal) mode of interaction, then group improvisation, where one or more group-members are computers, can be understood as a human-machine interaction. Human-computer group improvisation is then a human interaction with procedures, where the procedures are realised by a Universal Turing Machine.[13] In other words an improvisation, for example by a human-computer duo, is then an interaction between a human and a concrete instantiation of formal system.[14] Consequently, we might consider music improvised by humans and improvisation machines

---

[13] Current digital computers are examples of the Universal Turing Machine (Petzold, 2008).

[14] In his 1961 argument against understanding the mind as a machine, Lucas pointed out "it is of the essence of being a machine, that it should be a concrete instantiation of a formal system" (Lucas, 1961, p. 113).

together as being human-algorithm interactions; in effect these improvisations are interactions between human and mathematics; or we might say that these are human non-verbal 'dialogues' with an abstract and ideal world represented by mathematics which are expressed within the limits of Universal Turing Machines.

In human-computer group performance, the improvisation 'dialogue' is between humans and procedures that output computed numbers. By definition these numbers must be computable, where computable numbers are a subset of all numbers. Is this dialogue between human and Universal Turing Machine then a semantic exchange (is it an exchange of meaning) or is it perhaps purely syntactic (as abstract structures)?

The potential for 'semantic' qualities of the human-computer interactions seems to be a necessary condition for human-computer co-creativity. Without potential 'semantic' qualities it would seem that any 'creativity' of a human-computer system would be entirely due to human activity. In that case creativity would be one-sided, and the computer would be relegated to a tool-like position which might be interactive, but which would not constitute *an agent that contributes* to the co-creative results of the group performance system. As a 'tool' the computer could then be understood as an environmental extension of human cognition and creativity in the sense of Dror and Harnad[15] mentioned in Section 2.

However, we can say that the semantic contents

---

[15] Dror - Harnad (2008).

of a procedure may result from the transformation of input to output, whereas the semantic contents of human activity may come from the process of transformation of reference, where reference is enmeshed with memory and the intertextual network of the wider human culture.[16] In the case of music, semantic contents consist of sonic phenomena and concepts (when we consider musical 'structure' or 'form' as conceptual). So while human and computational semantic processes/contents are in different categories, the manifestations of outputs that are effected by these processes/contents may interact and such inter-category interactions may serve simultaneously as inputs or affects to both categories of semantic processes. Humans can listen and react to computer generated music, and some computer systems can extract features of human performances and use these features as inputs and/or parameters for music/sound generation.[17] The human experience of the sound as 'music' remains a human experience even if computational 'semantics' contribute to that experience.

6. Human creativity and formal systems

It seems unlikely that human creativity can be represented as a Universal Turing Machine with

---

[16] As in foot note 3, I use the notion of 'intertextual network' in the sense of Klein (2005).

[17] An example is documented in Mogensen (2020).

our current formal understanding of such machines. Gödel's Theorem indicates that all consistent formulations of number systems include some statements that cannot be proven.[18] So any consistent computational system will in this sense be incomplete, and this can also be interpreted as the system being unable to 'examine' itself. It is possible, perhaps even very likely, that humans are *not* consistent systems in computational terms, and that this might be considered a source of potential human 'creativity'. Human creativity may therefore be considered as a category that is distinct from any creativity based on a formal system. This may be true, at least in part, if we assume that humans are not 'computable' by a consistent number system. In other words, the mind is not reducible to a Universal Turing Machine. Given this assumption, human creativity cannot be described completely by computational models, nor do humans necessarily function as computational systems. So computational creativity must be something other than human creativity if we use a functional notion of creativity.

But if creativity is a product[19] then it would seem that the functional differences between human and computational system would not be decisive for creativity. A creativity product may be derived from experiential learning processes.[20] Experiential

---

[18] Gödel's Theorem in Hofstadter's paraphrase: "All consistent axiomatic formulations of number theory include undecidable propositions" (Hofstadter, 2000, p. 17).

[19] Glickman (1976).

[20] I use 'experiential learning process' in the sense of Dewey (1938) and Kolb (2015).

learning may be a necessary feature of human life, and computational systems may encode a functionality that is similar, at least in principle, by employing for example 'dynamic concept spaces'; and so a computational and dynamic concept space can form the basis for 'combinational creativity', 'exploratory creativity' and/or 'transformational creativity', in Boden's sense,[21] by computational systems.[22] In current machine learning techniques, what might be called 'experiential' learning is primarily an accumulation of 'memory', usually in the form of weights in artificial neural networks. This probably differs substantially from human experiential learning, which is likely to include embodied knowledge and musical imagination; these features, if they have computable parts, have yet to be modelled decisively in computational systems.

Creative 'musicking' [23] may be dependent on some form on knowing how to be musically creative, where the 'knowing how' refers to an active, and experientially learned, practice of musical creativity. This is distinct from a theory or model of creative musicking even if that theory or model could have explanatory powers about musical creativity. As Gilbert Ryle stated it: 'knowing how' is not reducible

---

[21] See Boden (2004).

[22] In some music improvising computer systems dynamic possibility spaces are parts of the design (Mogensen 2018, 2020). It would also seem that the particular features of musical output from a computational creativity could be claimed as defining the 'identity' of that creative agent, at least in part. This would be similar to the idea that the particular interpretive or improvisational style of a musician is defining of the musical identity of that musician.

[23] 'Musicking' as a verb in English is a term from Small (1998).

to 'knowing that'.[24] In both machine and human improvisation then, the capacity to contribute musically to a performance may be consistent with an inability to examine that performance from a theoretical perspective. The *knowing how* to contribute to a distributed creativity product, such as an improvised music performance, is a kind of creativity.

7. Knowing how to contribute

The 'knowing how' to contribute to improvised music implies a specific skill or capacity for making creative product(s) and versions of such skill may exist in a computer system. Gifford (*et al.*) proposed a taxonomy of computational music improvisation systems (such as Cypher, Voyager, Shimon, Omax, and others) but found that "the design of computational improvisers has been largely individual and ad-hoc"; improvising systems "are often designed for a specific improviser and performance style" and "the field is diverse, fragmented and lacks a coherent framework".[25]

The evaluation by Gifford (*et al.*) points to a dichotomy: the encoding of a formal system that 'knows how' to improvise music may be based on an explanatory theory (a framework), or it may be constructed *ad hoc*. Gifford (*et al.*) imply that the

---

[24] Ryle (2000), Ch. II.

[25] Gifford *et al.* (2018), pp. 19-20.

explanatory theory in the form of a 'framework' would be better, perhaps because the systems could thereby come closer to achieving human 'agency' in improvisation. I doubt the validity of this implication. Agency and explanatory theory may interact, however it seems that neither is a necessary prerequisite for the other in human activity; again, 'knowing how' is not reducible to theory. If this is true for humans then there is little reason to think that agency and theory should necessarily be co-dependent in computational systems. Having a 'coherent framework' to explain improvisation is not a necessary prerequisite for programming a computer improviser: the resulting computer improviser may 'know how' to contribute to improvisation, even if the potential contribution does not include a theoretic rationale.

We might question the applicability of the idea of a computer system 'knowing', even if it is capable of the 'how' of music improvisation. Ryle applied the 'knowing how' to human activity, not to computational 'activity', and perhaps 'knowing how' in the human sense implies some subjective experience of a capacity. Whether subjective experience could emerge from a sufficiently complex Universal Turing Machine is an open question. But the computational 'knowing how' would be a kind of 'trained computational system knowing how': the system consists of the software and hardware which is programmed, and the system is 'trained' in the sense that artificial neural networks are trained. The computational knowing is then 'embodied' in this system. This *computationally embodied knowing how* does not require some internal subjective experience to create sound that can be heard by humans as music.

8. Conclusion

My claim is that computational creativity applied to improvising music can be a contributor alongside human improvisational activity to a distributed group creativity product, which is music. The computational contribution to the creative musicking of the improvising group seems dependent on human cognitive involvement. The computational improvisation system does not necessarily model human improvisational activity and these should be considered categorically different. Also, the semantic capacities of humans and computational systems differ categorically, yet these categories allow interaction. A computationally embodied 'knowing how' to improvise may exist in a Universal Turing Machine without this giving a theoretical explanation of 'knowing why' in Ryle's sense. Such computationally embodied 'knowing how' to improvise is a kind of computational creativity which can contribute to the performance of improvised music.

I conclude that an individual 'agent' (human or computer) may contribute to the distributed creativity of a group improvisation. This distributed creativity may to be an emergent property of the group activity. The distributed creativity of the music performance group can be experienced as a creative product: the music. Distributed creativity may involve a process whereby the group members (the 'agents' of the performance system) interact in some ways to enable the emergence of that distributed creativity as evident in the music (the creative product). The purposefulness of human musicking is likely to be very different from the purposefulness that derives from programmed or 'trained' agency in a computer system. However, when engaged in group improvisation both kinds of purposefulness can be engaged in contributing to the music, and so result in co-creative music performance.

References

Boden, M. A. (2004): *The Creative Mind*, London: Routledge.

Brindle, R. S. (1967): *Serial Composition*, London: Oxford University Press.

Brindle, R. S. (1987): *The New Music*, London: Oxford University Press.

Bringsjord, S., Bello, P., & Ferrucci, D. (2000): *Creativity, the Turing Test, and the (Better) Lovelace Test*, «Minds and Machines», vol. 11, pp. 3-27.

Carnovalini, F., - Rodà, A. (2020): *Computational Creativity and Music Generation Systems: An Introduction to the State of the Art*, «Frontiers in Artificial Intelligence», vol. 3, issue 14, pp. 1-20.

Dewey, J. (1938): *Experience and Education*, New York, NY: Macmillan.

Dror, I. E., - Harnad, S. (2008): "Offloading Cognition onto Cognitive Technology", in I. E. Dror - S. Harnad (Eds.),*Cognition Distributed, How Cognitive Technology Extends our Minds*, Amsterdam: John Benjamins Publishing Company.

Gifford, T. - Knotts, S. - McCormack, J. - Kalonaris, S. - Yee-King, M. - d'Inverno, M. (2018, preprint): *Computational Systems for Music Improvisation*, «Digital Creativity», vol. 29, issue 1, pp. 19–36.

Glickman, J. (1976): "Creativity in the Arts" in L. Aagaard-Mogensen (Ed.), *Culture and Art*, Atlantic Highlands, NJ: Humanities Press.

Hofstadter, D. R. (2000): *Gödel, Escher, Bach: An Eternal Golden Braid*, London: Penguin Books.

Klein, M. L. (2005): *Intertextuality in Western Art Music*, Bloomington, IN: Indiana University Press.

Kolb, D. A. (2015): *Experiential Learning*, Upper Saddle River, NJ: Pearson Education, Inc.

Linson, A., - Clarke, E. F. (2017): "Distributed Cognition, Ecological Theory and Group Improvisation", in E. F. Clarke - M. Doffman (Eds.), *Distributed Creativity Collaboration and Improvisation in Contemporary Music*, New York, NY: Oxford University Press.

Lucas, J. R. (1961): *Minds, Machines and Gödel*, «Philosophy», vol. 36, issue 137, pp. 112–127.

Mogensen, R. (2018): "Dynamic Concept Spaces in Computational Creativity for Music", in V. C. Müller (Ed.), *Philosophy and Theory of Artificial Intelligence*, Cham: Springer.

Mogensen, R. (2020): "SomSwarm 1 Computational Creativity

Music Improvisation: Human-Computer Co-Creativity" in *Proceedings of the 5th international conference on live interfaces*, Trondheim: NTNU.

Petzold, C. (2008): *The Annotated Turing: A Guided Tour Through Alan Turing's Historic Paper on Computability and the Turing Machine*, Indianapolis: Wiley Publishing.

Ryle, G. (2000): *The concept of mind*, London: Penguin Books.

Small, C. (1998): *Musicking*, Middletown, CT: Wesleyan University Press.

Turing, A. (1936): *On Computable Numbers, With an Application to the Entscheidungsproblem*, «Proceedings of the London Mathematical Society», 2nd series, vol. 42, pp. 230–265.