SomSwarm 1 computational creativity music improvisation: human-computer *co-creativity*

René Mogensen [1]

[1]Royal Birmingham Conservatoire/Birmingham City University, Birmingham, UK, Rene.Mogensen@bcu.ac.uk

Abstract.

SomSwarm 1 is a 'computational creativity' implemented for musical improvisation performance together with human musicians, and I argue that the implementation is *co-creative* with human performers in concert music settings. *SomSwarm 1* is a first-generation hybrid system combining competitive self-organising maps with a 'swarm' algorithm: an artificial swarm moves in a dynamic topological space which is formed by the weights in self-organising networks that are trained with quantifications of features from human performances. The first concert applications of the implementation has been in two music contexts: *SomSwarm* with Big Band and *SomSwarm* in a duet with a human soloist. I analyse the implementation in relation to a formal specification for computational creativity in music and I examine the performance situation as a human-computer *co-creative* activity system. This analysis points to potential future developments for this kind of hybrid system and its 'co-creative' role as an improviser in musicking contexts.

Keywords.

Computational creativity, swarm algorithm, machine improvisation, machine learning, self-organising map, analysis.

Introduction

Previous analytical research using a formal specification for 'computational creativity' suggested a potential hybrid system which would combine Self-Organising Maps and Swarm algorithms (Mogensen, 2018a, pp. 9–12). The *SomSwarm 1* system is a first-generation network–swarm hybrid derived from some of the suggestions of the previous formal analysis, and was expressly implemented for music improvisation in human-computer *co-creative* group performances. In this hybrid technology an artificial swarm moves in a dynamic topological space which consists of the weights in self-organising networks that have been trained with quantifications of musical features from human performances; and so the networks take on a memory-like function which can be 'explored' by the swarm agents. The swarm activity is mapped onto electronic instruments and effects processing parameters which constitute the output of the improvising computer implementation.

To start I discuss a working understanding of computational creativity and human-computer 'co-creativity' which forms the basis for the subsequent exposition, and proceed to give an overview of a computational creativity specification previously developed. Some technical concepts of competition-based networks with swarm algorithms that form the key to the *SomSwarm* implementation architecture are introduced using the specification in Z-style notation; and some brief descriptions of implementation specifics give a sense of the implementation approach. As of this writing, *SomSwarm 1* has been active in two concert music contexts: first in *Inner Surfaces* for Big Band and *SomSwarm 1*; and secondly in *As a silence is interrupted* for *SomSwarm 1* and soprano saxophone. Both works are examined to give empirical support for the capacity for co-creativity in the first-generation *SomSwarm* technology.

A working understanding of 'computational creativity'

Creativity research has led to a variety of theories and definitions with no definitive description found yet and so 'we are still far from fully understanding what creativity really means' (de Sousa, 2008, 56). In common parlance artefacts and concepts made by humans are deemed 'creative' products even without having a clear view of what human creativity is, how it works, or whether it is computable; however, when hypothesising that it is possible, at least in principle, to make a digital computer-based 'computational creativity' then it must also be possible to make a *specification* of what a computational creativity *does* computationally, even if creativity might be an emergent result of a complex system. In other words, I take as given that anything that a current digital computer (or a Universal Turing Machine[1]) can do, can be represented in a formal specification. Therefore, if a computer can in some way be programmed to perform creative output, then it must be possible to define such a computer program in a formal specification of 'computational creativity'. Below I give an overview of such a specification and subsequently apply it as an analytical tool in order to compare implementations as well as to formulate ideas for developing new implementation possibilities.

I submit computational creativity and human creativity as logically distinct categories, both of which may be members of a 'creativity concept family'. In developing a working specification for computational creativity the vaguely defined 'human creativity' can serve heuristically as a prototype for the creativity concept family, but only in the sense that terms adopted from ideas about human creativity can be used to name and to guide the conceptualisations of components in the specification for computational creativity. Such use of terms does not imply any identity between human creativity and computational creativity.

I use a notion of *creativity* not as a process, but instead as a *product* (which echoes Glickman (1976)) of a *learning* process; for example an *experiential* learning process: in Kolb's (2015) interpretation of Dewey (1938) experiential learning is an iterative process with cycles of 'Impulse' leading to 'Observation' and to 'Knowledge' and 'Judgement' which result in what Dewey calls 'Purpose' (Mogensen, 2018b, 65–67). Such *purpose* is directed towards achieving or outputting particular *produced* artefacts and/or ideas. In order to formulate a specification for *production* by computational creativity, a necessary condition will be the inclusion of a learning process which may (or may not) result in creative outputs. In the recently developed *SomSwarm 1* implementations of computational music improvisers, such a learning process is interpreted as the process of building a 'memory' of patterns (in the form of self-organising maps) that is 'learned' through performance with humans, and which is explored as a possibility space by 'swarm agents'.

Human-computer 'co-creativity'

When we concede improvisation to be a creativity *product*, and allow that computer systems may 'co-improvise' with human performers, then the implication is that the computer systems are computationally 'creative'. Given *computational creativity* in a co-improvising relation with the human performers, the resulting human-machine 'co-creativity' is inclusive of human creativity and computational creativity, and so the relation between human and machine improvisers may be one of partnership, rather than one of user and tool.

In other words, co-creativity is an approach to human-computer relations which may transcend concepts of Human Computer Interface (HCI) as a human-centric user-tool paradigm giving hierarchical preference to the human and the goal-oriented activities of the human user of a computer system (Dix et al., 2007). Robert Rowe (1993) proposed a taxonomy which polarised what he called the 'instrumental paradigm' and the 'player paradigm' in computer interactivity. Using Rowe's categories the area of HCI is an 'instrumental paradigm' in the sense that the computer systems extend the human musical activities. On the other hand, human-computer *co-creativity* may be similar to Rowe's 'player paradigm' where the computer exhibits more independence as 'an artificial player, a musical presence with a personality and behavior of its own, though it may vary in the degree in which it follows the lead of the human player' (Rowe, 1993, 8). The 'human activity system', in the Soft Systems Analysis sense (Wilson, 2001), that

[1] The Universal Turing Machine as presented by Alan Turing (1936). 'The [Universal] Turing Machine not only established the basic requirements for effective calculability but also identified limits: No computer or programming language known today is more powerful than the Turing Machine' (Petzold, 2008, 330).

engages in improvisation, and which includes computational creativity as a co-improviser with humans, has a potential emergent creative performance output that is wider in scope than either purely human improvisation or purely computational improvisation systems; and such creative output is a result of human-computer co-creativity (Mogensen, 2019).

Overview of the *computational creativity* specification

I am developing a formal specification of computational creativity for music. The specification is a generalised model of computational creativity for music in the sense that it does not indicate specific technological solutions; the specification is concerned with *what* the computational creativity does, not *how* this would be implemented in software. An implemented system such as *SomSwarm 1* can then be analysed by identification of the functions of the system which correspond to components of the specification. Components of the specification include dynamic *Possibility Spaces* (Mogensen, 2018b), *Memory* and *Context* (Mogensen, 2020), as well as *Motivations* (Mogensen, 2017). Initial ideas for the specification were built on the earlier framework proposed by Wiggins (2006). Figure 1 gives a condensed and diagrammatic overview of the formal specification which indicates the more significant components necessary for the present discussion (where t represents discrete time).

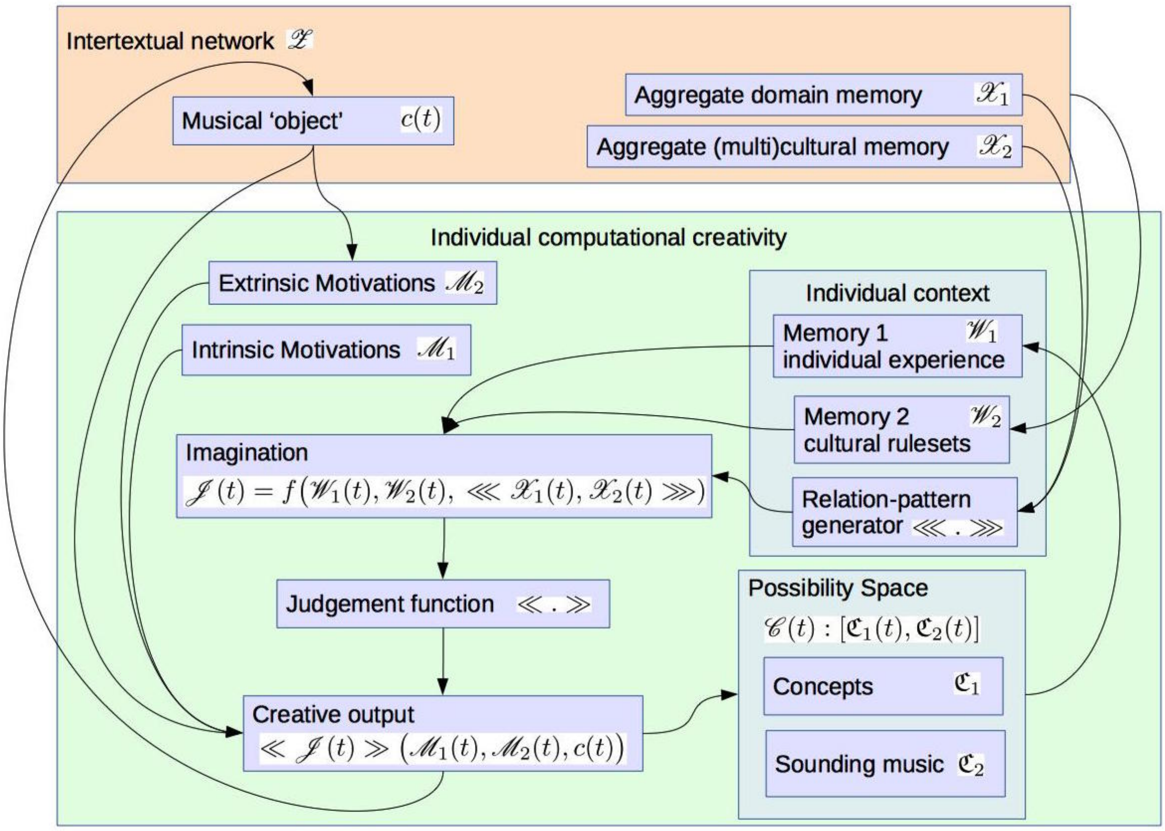

Figure 1: Diagram overview of the specification for computational creativity.

The individual computational creativity has memory which encompasses several categories of components: individual agent experience $\mathscr{W}_1$, cultural rulesets $\mathscr{W}_2$, aggregate domain memory $\mathscr{X}_1$ (idiomatic to styles or genres of music), and aggregate (multi)cultural memory $\mathscr{X}_2$ (larger cultural baggage). I proposed the *Individual context* $[\mathscr{W}_1, \mathscr{W}_2, \lll . \ggg]$ as well as the *Intertextual* network $\mathscr{Z}$ that enter the memory of the computational creativity as feeding an individual *Imagination* function $\mathscr{J}(t)$ (Mogensen, 2020). *Imagination* and *Motivations* $[\mathscr{M}_1, \mathscr{M}_2]$ enter into a *Judgement function* which shapes the possibility space and may result in a *Creative Output*. In the present context this *Creative Output* is improvised music. I have applied the formal specification in analyses of computational creativity implementation architectures, and the first version of the *SomSwarm* computational creativity was derived as a hybrid of two earlier implementation architectures. In the next section I turn to details of *SomSwarm 1*, with references to the specification outlined in Figure 1.

SomSwarm 1: competition-based network as topological space for swarm agents

The key concept in the present version of *SomSwarm 1* is the idea of a swarm algorithm which is active within a topological space, where the space is based on the weights of a self-organising map: these weights represents a kind of distributed 'memory' of the musical features of a performance, and the swarm agents moving around in this memory are then a second-order algorithm that superimposes, or nests, into the first-order weight network. The swarm agents can in this setting 'move within' the memory space of the network, reacting to the network weights as features of a topological space. In this system the swarm 'agents' which move according to simple rules may contribute to potentially 'creative' musical results when the movements and positioning of the agents are mapped to sound control mechanisms. This idea is outlined formally using Z-style notation in Figure 2 with references to the specification for computational creativity outlined in Figure 1.

SomSwarm 1

$t : \textit{time}$

$\mathscr{C}(t) : \textit{Possibility space } [\mathfrak{C}_1(t), \mathfrak{C}_2(t)]$

$N_i(t) : \textit{Network of weights } (\mathfrak{C}_2(t))$

$A_k(t) : \textit{Swarm Agent}(k) \textit{ position}$

$\mathscr{R}_k : \textit{Agent behaviour rule set}$

$\mathscr{C}_k(t) : \textit{Possibility Space visible to Agent}(k)$

$M(t) : \textit{Musician performance}(\textit{input})$

$c(t) : \textit{Musical object } (\textit{sounding music})$

$$M(t) \in c(t)$$
$$N_i(t) = f[N_i(t-1), c(t)]$$
$$\mathfrak{C}_2(t) = N_i(t)$$
$$\mathscr{C}(t) = [\mathfrak{C}_1(t), N_i(t)]$$
$$\forall\, k : \mathscr{C}_k(t) \in \mathscr{C}(t)$$
$$\forall\, k : \mathscr{C}_k(t) = \mathscr{R}_k[N_i(t)]$$
$$\forall\, k : A_k(t+1) = f_k\big(\mathscr{C}_k(t), \mathscr{R}_k, A_k(t)\big)$$
$$\mathscr{M}_1(t) = \bigcup_k \big[\mathscr{R}_k[N_i(t)]\big]$$
$$\mathscr{M}_2(t) = f[c(t)]$$

Figure 2: *SomSwarm 1* central idea: weight networks as topological space(s) providing the environment for swarm activity.

Briefly, in Figure 2, the network weights N_i encode the music phenomenon space $\mathfrak{C}_2$. At time t, the possibility space $\mathscr{C}(t)$ then consists of $\mathfrak{C}_2$ and a statically encoded concept space $\mathfrak{C}_1$. Each $Agent(k)$ in the group of k swarm agents can observe and move according to a ruleset $\mathscr{R}_k$ in the agent's visible possibility space $\mathscr{C}_k(t)$ which is a subset of the possibility space $\mathscr{C}(t)$. For each $Agent(k)$ the function f_k processes the visible possibility space $\mathscr{C}_k(t)$ according to its ruleset $\mathscr{R}_k$ and present location $A_k(t)$ to calculate its movement to the next position $A_k(t+1)$. Intrinsic motivations (at time t) of the system $\mathscr{M}_1(t)$ can be understood as the set of k agent rulesets $\mathscr{R}_k$ each of which is interacting with the network weights $N_i(t)$. Extrinsic motivations (at time t) of the system $\mathscr{M}_2(t)$ can be understood as a function of the current sounding music $c(t)$.

The self-organising network which contains the weights (or topological space) in *SomSwarm 1* is organised as a set of matrices. The set of matrices is cumulative and can be expanded so there is a time-based dimension to the memory where memory from earlier performance can affect later performances. The network weights are adjusted based on competition, as a version of a Kohonen Self-Organising Map (Fausett, 1994, 169–187). The inputs to the network are features of the performed music, primarily the sound of the human musician(s) captured through microphones. In the first implementations the feature extraction process has been focused on encoding pitch, rhythm, and attack intensity; but future work will expand this process to include more feature dimensions.

Because the network is competitively self-organising, where training continues during performance time, mapping is automated and the musical patterns mapped to the computer sound are based on characteristics of the network node weights: rhythmic dynamic patterns are generated from the weight value sets of clusters in the self-organising map. These patterns are associated with pitches derived mostly from a relatively short-term 'memory' matrix of pitch classes that were captured via one or more microphone inputs. The parameters generated by the swarm agents' positioning in the matrices are mapped to midi instruments, audio-file manipulations and 'gating' of audio processing.

Inner Surfaces for Big Band and *SomSwarm*

Inner Surfaces for Big Band and *SomSwarm* was premiered on September 3, 2019 in Aarhus Musikhus, Denmark by the ensemble *Blood Sweat Drum and Bass* under the direction of Jens Christian Chappe Jensen.[2] The work is encoded in a written score for the instrumentation of the ensemble and as a sequence on the computer which enables playing of several electroacoustic sound constructs at various points in the score as well as timed enabling and disabling of the *SomSwarm 1* improvising output in coordination with the score. The computer improviser is limited to three segments in the score; and the computer output is mapped to midi-instruments that have timbres that stand in contrast to the ensemble sounds. The three segments where *SomSwarm* plays there are also saxophone soloists improvising, and there is improvised interaction between *SomSwarm* and the soloists so that those segments add distinct ideas to the ensemble sound and to the shape of the musical work.

After rehearsals (September 1–3, 2019) I asked the saxophone soloists from the ensemble *Blood Sweat Drum and Bass* what their impressions were of interacting with the computer in the solo sections. One of them felt quite confident and said he found it fun to play with and interesting that the computer part was different every time the piece was played. The other saxophonist felt somewhat less confident about interacting with it but nevertheless came across in a musically convincing way during the rehearsals and concert performance.

The intention with *SomSwarm 1* in the work *Inner Surfaces* was to integrate a system that could be *co-creative* with human performers in an improvisational setting. The contribution of *SomSwarm 1* in this music work clearly challenged the soloists to improvise interactively with the computer sound. I propose that the resulting music was *co-created* by the human-computer ensemble. While the musical ideas played by *SomSwarm 1* were based on feature extractions from human performance, and technically could have been performed by humans, the machine generated music was unique to each rehearsal/performance and included both predictability and surprise within constraints demanded by the scored parts of the music. Therefore I argue that the parts 'improvised' by *SomSwarm 1* contributed co-creatively to the music.

As a silence is interrupted for soloist and *SomSwarm*

In the improvised duet entitled *As a silence is interrupted*[3] for *SomSwarm* and saxophone, the role of *SomSwarm 1* is more extensive than in the Big Band piece *Inner Surfaces.* The duet is used to develop the capabilities of *SomSwarm* further after the initial success of its part in *Inner Surfaces.* In the duet a structure for improvisation is planned before performances, through a sequence of mappings, so that the orchestration of the *SomSwarm* output changes in a predetermined sequence over the time of the performance. Structural points of the changes in mappings are activated mostly by the human performer who thereby controls the timing of these changes. Within the orchestration structure, *SomSwarm* plays

[2]As of this writing, a concert recordings of *Inner Surfaces* is available online at:
http://soundcloud/renemogensenmusic/20190903-innersurfaces

[3]As of this writing, a recordings of *As a silence is interrupted* is available online at:
http://soundcloud/renemogensenmusic/as-a-silence-is-interrupted-20190925

gestures with pitch and rhythmic qualities that are derived from the weights in the network; these weights are trained whenever the system is active.

In this version of *SomSwarm 1* the memory matrices from *Inner Surfaces* are also included as part of the topological space in which the swarm agents act and interact. I expect that with more performances and involvement with more music works, the *SomSwarm* memory network can be expanded, and will be limited only by hardware capacities. The first public concert performance of *As a silence is interrupted* was at the Royal Birmingham Conservatoire, UK, on November 6, 2019, with me on saxophone; rehearsals for this performance as well as the performance has added further training and implementation development has also enlarged the size of the memory network. Future plans call for performances with other soloists and ensembles.

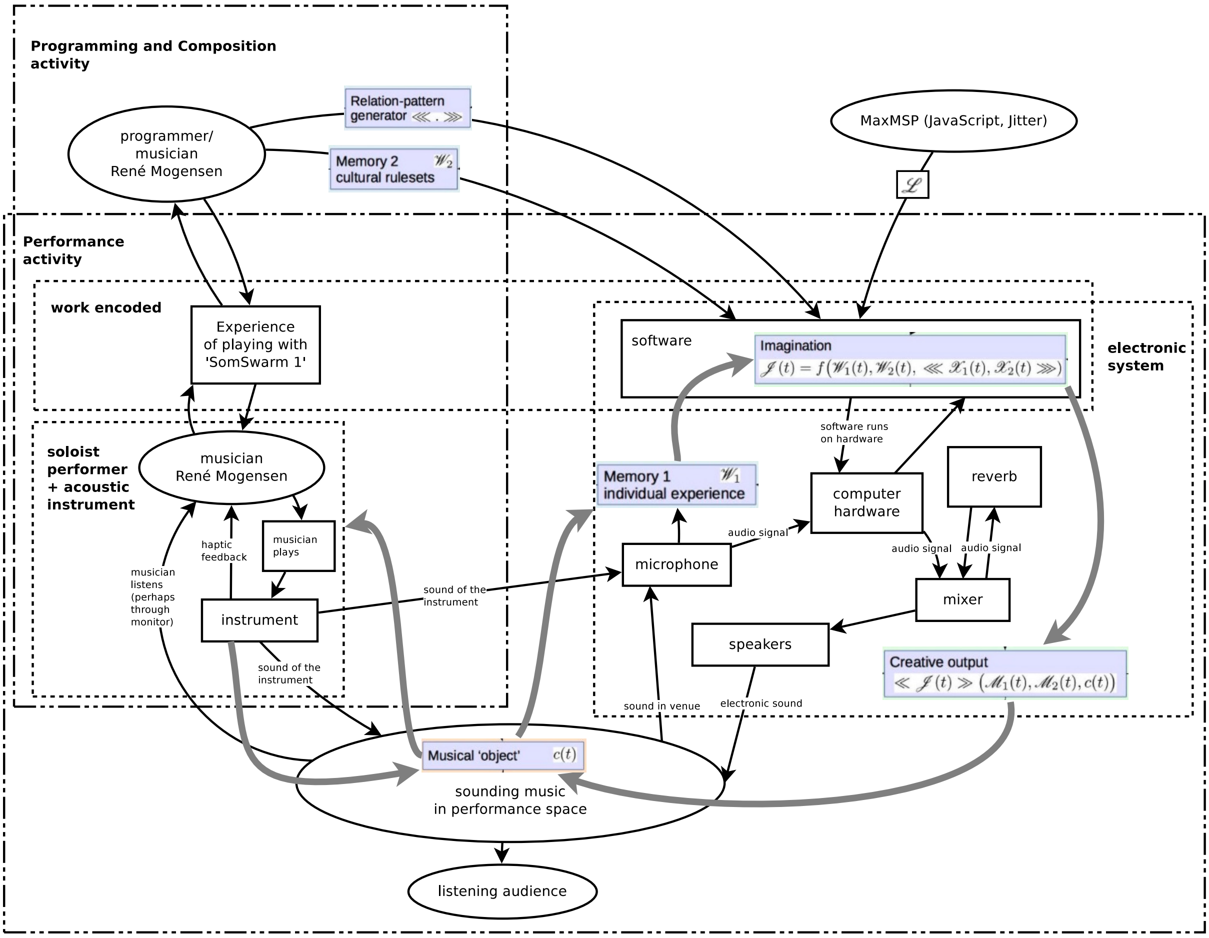

Figure 3: Soft Systems analysis of *As a silence is interrupted* for *SomSwarm* and saxophonist.

In Figure 3 a soft systems analysis of the performance situation of *As a silence is interrupted* shows both physical components and abstract specification components (boxes and/or ovals), and arrows indicate flows of influences and/or information. For now I will bypass a detailed narrative about this analysis, but the parallel and interacting development of two individual 'memories' suggest that it is a system that can result in *co-creativity*; by two individual 'memories' I refer to: 1. the programmer/musician gaining Experience of interacting with the system; and 2. the development of the *Memory 1* ($\mathscr{W}_1$) in the Computational Creativity *SomSwarm*. I suggest that these memories enable transformations of references in the human, and transformations of inputs to outputs in the software; and that capacities for these kinds of transformations are necessary for the respective music improvisation categories.

As was the case in the Big Band piece, the intention with *SomSwarm 1* in the work *As a silence is interrupted*, was to make a system that could be *co-creative* with a human performer in an improvisational setting. In my own performance interactions with the system I have found that training the system makes the musical output more interesting to me as an improviser; and my construction of the structure (sequence of mappings) grew out of improvising with the system. Also, I found that improvising with the system gradually made me adapt some of my musical choices to better interact with the output of the system: so an iterative process, of improvising and adjustment, provided learning for me as well as training for *SomSwarm 1*; I suggest understanding this as a development based on improvisation and interaction between me (as the programmer/improviser) and the software (as a computational creativity).

My own introspective impressions of improvising with the system are possibly biased; since I am both programmer and performer I have in-depth knowledge and influence in the software development. Lewis's *Voyager* system (Lewis, 2000) has been criticised as being 'a system that is designed by Lewis, for Lewis, and would not work for all musicians' (Magnusson, 2010, 46); and a naive evaluation of *SomSwarm 1* might raise a similar criticism given my introspective observations. However, the performances by *SomSwarm* in the *Inner Surfaces* Big Band setting, where I was not involved as a musician, gives some preliminary indications that the approach to implementing *SomSwarm* has some potential as a 'computational improviser' in a wider range of contexts. More work is needed to gather information about human improviser perspectives on how *SomSwarm* implementations interact in improvisational contexts. For this research direction I am planning a number of new work structures which will engage with various improvising soloists, and in parallel make further developments in the specification and implementations of future generations of *SomSwarm* as computational improvisers.

Comparative analysis of *SomSwarm 1* as a hybrid

SomSwarm can be aligned with the computational creativity specification and the resulting component representations are indicated in Table 1. I refer to Mogensen (2018a) for a more detailed narrative on the previous comparative analysis of my *Favoleggiatori 2* and *MASOM* by Tatar and Pasquier (2017); here the focus is on the comparative analysis of *SomSwarm 1* and its qualities as a hybrid of the two previous implementation architectures.

Reading Table 1 we can observe some similarities and some differences between the three implementations: the most obvious similarities being that all three have static concept spaces $\mathfrak{C}_1(t)$, as well as implicit and constant cultural memory $\mathscr{W}_2(t)$ which implies a somewhat static mapping of computation output to sound. Differences in the phenomenon spaces $\mathfrak{C}_2(t)$ are therefore central to differences in the implementations' memories of possibility spaces $\mathscr{W}_1(t)$, and are significant: in *Favoleggiatori 2* there is a performance time memory matrix $\mathscr{W}_1(t)$ of pitch and rhythmic features, whereas in *MASOM* the memory $\mathscr{W}_1(t)$ is trained before performance time ($MASOM_T$ training $(t < 0)$). In *SomSwarm 1* the $\mathscr{W}_1(t)$ memory can be divided into three categories $N_i : [N_0, N_1, N_2]$ where N_0 is a short-term memory version from *Favoleggiatori 2*, N_1 is self-organising network that is trained during performance time, and N_2 is a collection of self-organising networks that have been trained in previous performances/rehearsals and may represent a longer-term memory of phenomenon spaces that have been input (as 'training') in the past. So in *SomSwarm 1* this multiple category $\mathscr{W}_1(t)$ has characteristics that resemble the $\mathscr{W}_1(t)$ functions in both *Favoleggiatori 2* and *MASOM*.

I have discussed (2020, 2018a) the potential significance of memory attenuation as a factor in computational creativity, and in *Favoleggiatori 2* the memory matrix $\mathscr{W}_1(t)$ is degraded gradually at a constant rate Q_1, whereas in *MASOM* there does not seem to be any explicit Q_1; in *SomSwarm 1* the three categories $N_i : [N_0, N_1, N_2]$ have differing attenuation and growth characteristics. The specification *Imagination* function $\mathscr{I}(t)$ has varying characteristics over the three implementations as a result of the different possibility space representations: in *Favoleggiatori 2* it is dynamic $N_0(t)$, *MASOM* with a static $MASOM_T(t)$, and *SomSwarm 1* with a dynamic and cumulative $N_i : [N_0, N_1, N_2]$. Finally, the *Intrinsic Motivations* in *SomSwarm 1* and *Favoleggiatori 2* are intertwined with agent behaviour rules $\mathscr{R}_k$ which are constant in these systems, whereas the (constant) *MASOM* strategy is to use random choices within a node classification of the self-organising map. In summary: *SomSwarm 1* includes components that are similar to components found in both *Favoleggiatori 2* and *MASOM* and may be considered a hybrid which I propose as being potentially more flexible, responsive, and adaptable than either of its predecessors.

Specification component	***Favoleggiatori 2***	***MASOM***	***SomSwarm 1***
$\mathfrak{C}_1(t)$: concept space	static	static	static
$\mathfrak{C}_2(t)$: phenomenon space	$\mathfrak{C}_2(t) = \mathfrak{C}_2(t-1) \cdot f_{\mathfrak{C}_2}\big(c(t), \mathfrak{C}_2(t-1)\}$	$MASOM_T$ constant ($t \geq 0$) $MASOM_T$ training ($t < 0$)	$N_i(t) = f[N_i(t-1), M(t), c(t)]$ ongoing training ($t > 0$)
$\mathscr{W}_1(t)$: memory of possibility space $[\mathfrak{C}_1(t), \mathfrak{C}_2(t)]$	$N_0(t) : \bigcup_{p=1}^{t-1} \Big(\mathscr{C}_F(p) \cdot Q_1\Big)$	$MASOM_T$ constant ($t \geq 0$) represented by SOM	represented by $N_i(t) : [N_0(t), N_1(t), N_2]$
$\mathscr{W}_2(t)$: cultural memory	S_3: implicit and constant	T_1: implicit and constant	S_4: implicit and constant
Q_1 : attenuation of $\mathscr{W}_1$	N_0 degrades at constant rate ($t > 0$)	none	N_0 degrades at varying rates N_1 learn rate attenuates ($t > 0$) N_2 increases outside time t
Q_2 : attenuation of $\mathscr{W}_2$	none	none	none
$\mathscr{I}(t)$: Imagination	$f\big(N_0(t), S_3, \lll N_0(t), S_3 \ggg \big)$	$f\big(MASOM_T, T_1, \lll MASOM_T, T_1 \ggg \big)$	$f\big(N_i(t), S_4, \lll N_i(t), S_4 \ggg \big)$
$\ll . \gg$: *Judgement* function	$\ll . \gg_{agent}$ agent behaviour rules	$\ll . \gg_{VOMM}$ VOMM → random selection from cluster	$\ll . \gg_{agent}$ agent behaviour rules
$\mathscr{M}_1$: *Intrinsic Motivation*	agent behaviour rules	$MASOM_{GR}$ Random audio from SOM node	agent behaviour rules
$\mathscr{M}_2$: *Extrinsic Motivation*	feature analysis	$MASOM_{GR}$ feature analysis	feature analysis

Table 1: Comparison chart of specification components in *Favoleggiatori 2*, *MASOM*, and *SomSwarm 1*, where performance time is represented by discrete time t.

Conclusion

Analytical use of the computational creativity specification outlined in Figure 1, in conjunction with the experiments of integrating the *SomSwarm 1* implementation as an improviser in *Inner Surfaces* and *As a silence is interrupted* have arguably resulted in satisfying music as creativity products. The musical challenges and surprises that create rich interactions for human improvisers in the two concert works mean that *SomSwarm 1* contributes to the creative product and I argue that it is potentially *co-creative* with the human performers. The analytical work related to *SomSwarm 1* has also given rise to ideas towards future *SomSwarm* versions for deeper co-creativity with human performers. However, further discussions about these possibilities await future work.

References

de Sousa, F. C. (2008). Still the elusive definition of creativity. *International Journal of Psychology 2*, 55–82.

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

Dix, A., J. E. Finlay, G. D. Abowd, and R. Beale (2007). *Human-Computer Interaction.* http://www.myilibrary.com?ID=106454 (accessed October 9, 2015): Prentice Hall.

Fausett, L. (1994). *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications.* Upper Saddle River, NJ: Prentice Hall, Inc.

Glickman, J. (1976). Creativity in the arts. In L. Aagaard-Mogensen (Ed.), *Culture and Art*, pp. 130–146. Atlantic Highlands, NJ: Humanities Press.

Kolb, D. A. (2015). *Experiential Learning* (2nd ed.). Pearson Education, Inc.

Lewis, G. E. (2000). Too many notes: Computers, complexity and culture in 'Voyager'. *Leonardo Music Journal 10*, 33–39.

Magnusson, T. (2010). An epistemic dimension space for musical devices. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, Sydney, pp. 43–46. NIME 2010.

Mogensen, R. (2017). Computational motivation for computational creativity in improvised music. In *Proceedings of Computer Simulation of Musical Creativity conference 2017*, Paper 11, https://csmc2017.wordpress.com/ proceedings/, pp. 1–10. Open University.

Mogensen, R. (2018a). Computational creativity specification for music: comparative analysis of machine improvisers *Favoleggiatori 2* and *MASOM.* In *Proceedings of Computer Simulation of Musical Creativity conference 2018*, https://csmc2018.wordpress.com/ proceedings/, pp. 1 – 15. University College Dublin.

Mogensen, R. (2018b). Dynamic concept spaces in computational creativity for music. In V. C. Müller (Ed.), *Philosophy and Theory of Artificial Intelligence 2017*, SAPERE, Berlin, pp. 57–68. Springer.

Mogensen, R. (2019). Human-machine improvisation: an analytical approach to human-computer co-creativity in improvised music. In *Proceedings of Computational Creativity Symposium at AISB Convention 2019.* Falmouth University.

Mogensen, R. (2020). Formal representation of context in computational creativity for music. In S. S. Gouveia and J. de Fernandes Teixeira (Eds.), *Artificial Intelligence: an Exploration*, Wilmington. Vernon Press (forthcoming).

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

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

Tatar, K. and P. Pasquier (2017). Masom: A musical agent architecture based on self-organizing maps, affective computing, and variable markov models. In P. Pasquier, O. Bown, and A. Eigenfeldt (Eds.), *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, www.musicalmetacreation.org/proceedings/mume-2017/. MUME 2017.

Turing, A. (1936). On computable numbers, with and application to the entscheidungsproblem. *Proceedings of the London Mathematical Society, 2nd series 42*, 230–265.

Wiggins, G. A. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems 19*(7), 449–458.

Wilson, B. (2001). *Soft Systems Methodology.* Chichester: John Wiley and Sons, Ltd.