Computational creativity specification for music: comparative analysis of machine improvisers

Favoleggiatori 2 and MASOM

René Mogensen
Birmingham Conservatoire/Birmingham City University, UK
Rene.Mogensen@bcu.ac.uk

Abstract. I introduce my machine music improviser entitled Favoleggiatori 2 which is an implementation based on swarm algorithms that operate in dynamic topological spaces, where these topological spaces represent memory of musical events. I align Favoleggiatori 2 with a formal specification for computational creativity which I have developed previously. The formal specification is used as an analytical tool to understand the limitations of the Favoleggiatori 2 architecture. This analysis based on the specification also facilitates comparative study of the MASOM machine improviser architecture in order to constructively formulate some of the differences and similarities of these architectures. The limitations indicated by the analytical use of the formal specification suggest a hybrid implementation architecture which is potentially more flexible and more musically adaptable than the two implementations analysed. This analytical approach can be useful in theoretical and technical development towards the next generation computational creativities with greater ‘creative’ potentials than current implementations.

Keywords: computational creativity, computational improvisation, improvised music, memory representation as topological space, analysis

1 Introduction

I have previously developed a formal specification for computational creativity for music in [9] [10] [11] and [12]. I propose that this specification is useful as an analytical framework for computational creativity and for developing potential future capabilities of computational creativity. I give an overview of this specification as a working model of computational creativity in section 2. I then align my specification with two implementations: my own recently developed Favoleggiatori 2 in section 3 and the MASOM system by Tatar and Pasquier [13] in section 4. These two implementations, or software architectures, are based on diverse technologies such as swarm algorithms and self-organising maps. The alignment of technologies with specification components facilitates comparison of diverse technologies via their functionality as represented in the specification. In section 5 I present comparative analysis using the specification as a reference, or bench-mark. As a result of the analysis I suggest a hybrid architecture for a
next-generation computational creativity which potentially can have more flexible musical responsiveness than the two examined systems. I argue that this analytical approach can help develop the future capabilities of computational creativity.

2 Overview of the specification: a working model of computational creativity

In [11] I proposed to understand creativity not as a process itself, but instead as a product (echoing Glickman [5]) of a learning process; I aligned my computational creativity model with an experiential learning model (e.g. Kolb’s [8] interpretation of Dewey [3]). The experiential learning process results in what Dewey calls ‘purpose’. I take such purpose to be directed towards achieving particular results or effects, and this is then the basis for an interpretation of creativity as evident in a produced artefact or idea. In order to formulate a specification for production, in this model of computational creativity, a necessary (but probably not sufficient) condition will be the enabling of a learning process which may (or may not) result in creative output. The formal specification that I am developing includes dynamic possibility spaces [11], memory [12], motivations [9] and is built on my critical reworking and expansion of the earlier framework proposed by Wiggins [15].

Fig. 1: Diagram overview of the specification for computational creativity.
In [12] I developed the specification to include an *Imagination* function that operates on *Memory* and I expanded the representation of memory to encompass two categories: 1. individual agent experience and 2. cultural rule sets. With these developments of the formal model I proposed the context of the computational creativity as being an individual *Imagination* function as well as the *Intertextual Network* that enter the memory of the computational creativity. An informal overview of the components of the model is given in Figure 1 [12], and formal specification schemata of selected components (in Z-style notation) are shown in sections 3 and 4. The *Relation-patterns* in this specification are results of pattern recognition functions applied to memory contents; *Imagination* and *Motivations* enter into a creative judgement function which shapes the possibility space and may result in *Creative Output*.

Given this specification as a model for computational creativity, I argue that using it as an analytical tool can stimulate development of implementations. In [10] and [9] I aligned earlier and simpler versions of the specification to two implementations: *Voyager* by George E. Lewis and my own *Favola*. In this article I align the now richer specification with *Favoleggiatori 2* and *MASOM* with a focus on the specification components *Imagination*, *Possibility Space*, and *Creative Output*.

3 Analysis: *Favoleggiatori 2*

*Favoleggiatori 2* is a work that includes ‘computational creativity’ as the basis for a machine improviser. The machine has been implemented with the intention that it should co-improvise together with a human instrumentalist and as such may be considered part of a duo of human and artificial musical intelligence. I argue that the machine and human are *co-creative* and that the *co-creative product* is the musical performance. The computational creativity may be considered a *partial creativity* in a music performance system.

The structure of the *Favoleggiatori 2* machine improviser includes dynamic topological spaces as representations of memory of musical activity in performance-time. More specifically, these representations in *Favoleggiatori 2* include three separate six-dimensional topological spaces, each with pitch in three dimensions and rhythm in three dimensions, that represent memories of co-performed musical activity. The topological spaces are explored by ‘swarms’ of artificial ‘agents’ and the topological features of the spaces are changed by feature analysis of the human audio performance in performance-time. The swarm agent activities

---

1 The specification for computational creativity is here expressed in Z-style notation [4]. Z is a specification language created to model computational processes without specifying coding solutions; I use this notation style for its conciseness and also with the aim of developing a practical specification of computational creativity for music.

2 See [10] for a more detailed discussion of an analytical view of *partial creativity* and *co-creativity* in a music performance situation.

3 The feature analysis is focused on the human instrumental performance, but features of the machine performance can also affect the topological spaces.
\[ J(t) \{ \text{Favoleggiatori 2} \} : \text{Imagination} \]

\[ \mathcal{L} : \text{Language } \{ \text{MaxMSP, JavaScript} \} \]

\[ \mathcal{R} : \text{Constraints on concept space } \{ \text{implicit, constant ; } S_1 \} \]

\[ \mathcal{F} : \text{Search strategy } \{ \text{swarm agent; movements in topologies} ; S_2 \} \]

\[ \mathcal{E} : \text{Value definition } \{ \text{implicit and constant} \} \]

\[ \mathcal{W}_1 : \text{Possibility space memory } \{ \text{topologies} \} \]

\[ \mathcal{W}_2 : \text{Cultural memory } \{ \text{implicit, constant ; } S_1 \} \]

\[ \mathcal{Q}_1, \mathcal{Q}_2 : \text{memory attenuation } \{ \mathcal{Q}_1 : \text{constant rate}; \mathcal{Q}_2 : \text{none} \} \]

\[ \mathcal{A}_1 : \text{Aggregate agent domains } \{ \text{constant ; } S_1 \} \]

\[ \mathcal{A}_2 : \text{Aggregate (multi)cultural rulesets } \{ \text{constant ; } S_1 \} \]

\[ \ll : \gg : \text{relation—pattern generator function } \{ \text{constant mapping} \} \]

\[ t : \text{Time} \]

\[ \mathcal{R}, \mathcal{F}, \mathcal{E} \in \mathcal{R} \]

\[ \mathcal{W}_1(t) = \bigcup_{p=1}^{n-1} \left( \mathcal{E}(p) \cdot Q_1(p) \right) \Rightarrow \bigcup_{p=1}^{n-1} \left( \mathcal{E}(p) \cdot Q_1 \right) \]

\[ \mathcal{W}_2(t) = \bigcup_{i=1}^{p-1} \left( (\mathcal{R}(i), \mathcal{F}(i), \mathcal{E}(i)) \cdot Q_2(i) \right) \Rightarrow S_3 \]

\[ \mathcal{A}_1(t) = \bigcup_{d=1}^{n} \left( \mathcal{W}_1(d, t) \right) \Rightarrow \mathcal{W}_1(t) \]

\[ \mathcal{A}_2(t) = \bigcup_{c=1}^{p} \left( \mathcal{W}_2(c, t) \right) \Rightarrow \mathcal{W}_2(t) \]

\[ J(t) = f(\mathcal{W}_1(t), \mathcal{W}_2(t), \ll, \mathcal{A}_1(t), \mathcal{A}_2(t) \gg \Rightarrow \Rightarrow) \Rightarrow f(\mathcal{W}_1(t), S_3, \ll, \mathcal{W}_1(t), S_3 \gg \Rightarrow) \]

Fig. 2: Favoleggiatori 2 is aligned in \{ \} parentheses with the declarations of the Imagination specification, and the predicates specific to the alignment follow after the ⇒ symbols.

\[ \mathcal{E}(t) \{ \text{Favoleggiatori 2} \} : \text{Possibility Space} \]

\[ \mathcal{E}_1(t) : \text{Concept space } \{ \text{constant ; } S_1 \} \]

\[ \mathcal{E}_2(t) : \text{Sonic (music) phenomenon space } \{ \text{agent potential moves at time } t \text{ in topologies} \} \]

\[ c(t) : \text{Instance of } [\mathcal{E}_1, \mathcal{E}_2] \]

\[ t : \text{Time} \]

\[ \Delta \mathcal{R}(t) = f_{\mathcal{R}}(\mathcal{R}(t-1), \mathcal{R}_2(t-1)) \Rightarrow 0 \]

\[ \Delta \mathcal{F}(t) = f_{\mathcal{F}}(\mathcal{F}(t-1), \mathcal{W}_1(t-1), \mathcal{W}_2(t-1), \mathcal{A}_1(t-1), \mathcal{A}_2(t-1)) \Rightarrow 0 \]

\[ \Delta \mathcal{E}(t) = f_{\mathcal{E}}(\mathcal{E}(t-1), \mathcal{W}_1(t-1), \mathcal{W}_2(t-1), \mathcal{A}_1(t-1), \mathcal{A}_2(t-1)) \Rightarrow 0 \]

\[ \Delta \mathcal{A}_1(t) = f_{\mathcal{A}_1}(c(t), \mathcal{E}_1(t-1), \Delta \mathcal{R}(t), \Delta \mathcal{F}(t), \Delta \mathcal{E}(t)) \Rightarrow 0 \]

\[ \Delta \mathcal{A}_2(t) = f_{\mathcal{A}_2}(c(t), \mathcal{E}_2(t-1), \Delta \mathcal{R}(t), \Delta \mathcal{F}(t), \Delta \mathcal{E}(t)) \Rightarrow f_{\mathcal{A}_2}(c(t), \mathcal{E}_2(t-1)) \]

\[ \mathcal{E}_1(t) = \mathcal{E}_1(t-1) \cdot \Delta \mathcal{E}_1(t) \Rightarrow \mathcal{E}_1(t-1) = \mathcal{E}_1 \]

\[ \mathcal{E}_2(t) = \mathcal{E}_2(t-1) \cdot \Delta \mathcal{E}_2(t) \Rightarrow \mathcal{E}_2(t-1) \cdot f_{\mathcal{A}_2}(c(t), \mathcal{E}_2(t-1)) \]

\[ \mathcal{E}(t) : [\mathcal{E}_1(t), \mathcal{E}_2(t)] \]

Fig. 3: The possibility space of Favoleggiatori 2 aligned with the specification.
within the topological space are mapped onto electronic synthesis, midi instruments and spatialisation parameters which constitute the output of the machine improviser.

I have aligned the structures of the implementation with the Imagination specification in Figure 2. The topological space dimensions are dimensions of the individual possibility space memory $W_1$. The pitch space representation is my adaptation of a two-voice organisation of pitch space from Tymoczko [14, pp. 73–79]. The rhythm possibility representation is a low-level mapping of rhythmic aspects of the human co-performer’s sound onto the topological space.

Concept spaces are fixed in programming by the mapping of human performance parameters onto topological features and the mapping of agent behaviour onto the shaping of musical events. As I align components of Favoleggiatori 2 with the Imagination specification it becomes evident that many components are constants and the alignment reduces the specification expressions as shown in Figure 2: $J(t)\{\text{Favoleggiatori 2}\}$ is a function of the $W_1$ possibility space represented by topologies with the cultural ruleset as a constant.

The Creative Output of Favoleggiatori 2 as shown in Figure 4 is based on a Judgement function which is shaped by Imagination $J(t)$ and uses Extrinsic Motivations $M_2$, Intrinsic Motivations $M_1$, and the current sonic event $c(t)$ as input. Intrinsic Motivations are rule-based in Favoleggiatori 2: three groups of six agents move within three separate six-dimensional topologies with swarm-like behaviour rules – these rules represent intrinsic motivations. Extrinsic motivations are fixed in Favoleggiatori 2: topologies are shaped by analysis of human performance parameters (pitch and rhythm).

<table>
<thead>
<tr>
<th>Creative Output {Favoleggiatori 2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{C}$ : Possibility space</td>
</tr>
<tr>
<td>$c(t)$ : Current musical object – instance of $\mathcal{C}{\text{applied to } W_1(t)}$</td>
</tr>
<tr>
<td>$M_1$ : Intrinsic Motivation {agent behaviour rules}</td>
</tr>
<tr>
<td>$M_2$ : Extrinsic Motivation {analysis of human performance}</td>
</tr>
<tr>
<td>$t$ : Time</td>
</tr>
<tr>
<td>$\ll , \gg $: Judgement function</td>
</tr>
</tbody>
</table>

$$\ll J(t) \gg (M_1(t), M_2(t), c(t)) \Rightarrow \ll f(W_1(t), S_3) \gg (M_1(t), M_2(t))$$

Fig. 4: Favoleggiatori 2 aligned with the Creative Output formalisation.

Some interesting outcomes of this analysis are the components shown to be constant in the implementation: as mentioned earlier, the cultural ruleset $W_2(t)$ and its aggregate $X_2(t)$ reduce to a constant $S_3$, so that the Imagination, as $J(t)\{\text{Favoleggiatori 2}\}$, is reduced to a function of possibility space memory $W_1(t)$ and the constant $S_3$. Furthermore, the Creative Output $\{\text{Favoleggiatori 2}\}$ is then a function of possibility space memory $W_1(t)$, which represents the dynamic topologies, and the constant ruleset $S_3$ which is a structure implicitly fixed...
in the implementation code. The *Judgement* function takes two additional inputs: 1. *Intrinsic motivation* $M_1$ which is expressed in the behaviour rules which govern the movements which the swarm agents make within the topologies; 2. *Extrinsic motivations* $M_2$ which are articulated by the mapping of human musical performance features onto the dynamic topologies. The topologies are dynamic in that they change over time. Another point to note is that in Favoleggiatori 2 there is an overlap between *Intrinsic Motivation* $M_1$ and the *Judgement* function $<$ since both express the agent behaviour rules. Some of the limitations of the implementation components can be surpassed with solutions suggested by comparative analysis of a different computational creativity architecture. In the following section I will apply a specification-based analysis to the *MASOM* machine improvisation implementation [13] to allow some comparative observations and suggestions for a hybrid implementation architecture in section 5.

### 4 Analysis: *MASOM*

Tatar and Pasquier describe their architecture for ‘Musical Agents based on Self-Organising Maps (*MASOM*’) as ‘a machine improvisation software for live performance’ [13, p.1]. They train a Self-Organising Map (SOM) with segmented corpuses of electroacoustic music. They use a Variable-Order Markov Model (VOMM) to choose a node in the SOM that is calculated to be the closest to the features of a segment of current improvisation encountered during a group improvisation including one or more human musicians. A random selection of an audio segment (of the training corpus) from the chosen SOM node is then played as a contribution to the group improvisation. An alignment of the architecture with my formal specification for computational creativity gives a review of the architecture.

\[
\text{MASOM} \\
\text{MASOM}_T : \text{Training (SOM, VOMM)} \\
\text{MASOM}_{GL} : \text{Generation (Machine Listening to Audio Input)} \\
\text{MASOM}_{GR} : \text{Generation (Random audio from SOM node)} \\
\text{MASOM Output} = f(\text{MASOM}_{GL} \rightarrow \text{MASOM}_T \rightarrow \text{MASOM}_{GR})
\]

Fig. 5: Summary of the MASOM components, from my interpretation of the published description [13].

The main components of *MASOM* are summarised in Figure 5: input audio features are analysed to find the closest SOM node from the training set, and the VOMM selects the SOM node that is calculated to be closest to the feature analysis results. This selection triggers a random choice of some element of the SOM node, in other words a random playback of some segment of the audio...
used for training, where the random selection is constrained to a subset of the training repertoire represented by the SOM node.

\[
\mathcal{J}(t) \{ \text{MASOM} \} : \text{Imagination}
\]
\[
\mathcal{L} : \text{Language} \{ \text{MaxMSP} \}
\]
\[
\mathcal{R} : \text{Constraints on possibility space} \{ \text{MASOM}_T \}
\]
\[
\mathcal{S} : \text{Search strategy} \{ \text{MASOM}_{GL} \rightarrow \text{MASOM}_T \rightarrow \text{MASOM}_{GR} \}
\]
\[
\mathcal{E} : \text{Value definition} \{ \text{implicit and constant} \}
\]
\[
\mathcal{W}_1 : \text{Possibility space memory} \{ \text{MASOM}_T \}
\]
\[
\mathcal{W}_2 : \text{Cultural memory} \{ \text{implicit and constant} : T_1 \}
\]
\[
Q_1, Q_2 : \text{memory attenuation} \{ \text{none} \}
\]
\[
\mathcal{X}_1 : \text{Aggregate agent domains} \{ \text{constant} \}
\]
\[
\mathcal{X}_2 : \text{Aggregate multi-cultural rulesets} \{ \text{constant} \}
\]
\[
\ll : \gg : \text{relation-pattern generator function}
\]
\[
t : \text{time}
\]

\[
\mathcal{R}, \mathcal{S} \in \mathcal{L}
\]
\[
\mathcal{W}_1(t) = \bigcup_{n=1}^{t-1} \left( \mathcal{C}(p) \cdot Q_1(p) \right) \Rightarrow \text{MASOM}_T
\]
\[
\mathcal{W}_2(t) = \bigcup_{n=1}^{t-1} \left( (\mathcal{R}(i), \mathcal{S}(i), \mathcal{E}(i)) \cdot Q_2(i) \right) \Rightarrow T_1
\]
\[
\mathcal{X}_1(t) = \bigcup_{n=1}^{t} (\mathcal{W}_1(d, t)) \Rightarrow \mathcal{W}_1(t)
\]
\[
\mathcal{X}_2(t) = \bigcup_{n=1}^{t} (\mathcal{W}_2(c, t)) \Rightarrow \mathcal{W}_2(t)
\]
\[
\mathcal{J}(t) = f(\mathcal{W}_1(t), \mathcal{W}_2(t), \ll \mathcal{X}_1(t), \mathcal{X}_2(t) \gg) \Rightarrow f(\mathcal{W}_1(t), T_1 \ll \mathcal{W}_1(t), T_1 \gg)
\]

Fig. 6: MASOM aligned with the Imagination specification.

The components of the MASOM architecture can then be aligned with the specification for comparative study. To this end I align the MASOM components with the same specification components used in the analysis of Favoleggiatori 2 in section 3. The Imagination specification aligned with the MASOM architecture is shown in Figure 6 with this architecture the \( \mathcal{J}(t) \) component is a function of \( \text{MASOM}_T \) and the constant cultural ruleset \( T_1 \).

As shown in Figure 7 the Possibility Space of MASOM is static, in comparison with the specification which allows dynamic possibility spaces (as discussed in 11). The specification Imagination component includes dynamic memories at the domain-level possibility space as well as at the cultural level, while the MASOM architecture is completely static in both of these areas. So for both functions \( \mathcal{C}(t) \) and \( \mathcal{J}(t) \) the MASOM architecture is significantly reduced in dynamic possibilities compared with the specification components. This can be interpreted as that the MASOM architecture explores a predetermined possibility space and has a static decision process. The alignment and comparison of the MASOM architecture with the framework specification is not intended to be prescriptive, rather it may indicate further and/or alternative possibilities for computational creativity architectures by giving some indications of the MASOM architecture limitations.
\text{\textit{C}(t)} \{\text{MASOM}\}: \text{Possibility Space}

\begin{align*}
\mathcal{C}_1(t) & : \text{Concept space} \{\text{constant}\} \\
\mathcal{C}_2(t) & : \text{Sonic (music) phenomenon space} \{\text{MASOM}_T = \text{Training (SOM, VOMM)}\} \\
c(t) & : \text{Instance of } [\mathcal{C}_1, \mathcal{C}_2] \\
t & : \text{Time}
\end{align*}

\begin{align*}
\Delta \mathcal{R}(t) &= f_\mathcal{R}(\mathcal{R}(t-1), \mathcal{I}_2(t-1)) \Rightarrow 0 \\
\Delta \mathcal{I}(t) &= f_\mathcal{I}(\mathcal{I}(t-1), \mathcal{I}_2(t-1), \mathcal{M}_1(t-1), \mathcal{M}_2(t-1)) \Rightarrow 0 \\
\Delta \mathcal{E}(t) &= f_\mathcal{E}(\mathcal{E}(t-1), \mathcal{I}_1(t-1), \mathcal{I}_2(t-1), \mathcal{M}_1(t-1), \mathcal{M}_2(t-1)) \Rightarrow 0 \\
\Delta \mathcal{C}_1(t) &= \mathcal{C}_1(t) - \mathcal{C}_1(t-1) + \Delta \mathcal{C}_1(t) \Rightarrow \mathcal{C}_1(t-1) \\
\Delta \mathcal{C}_2(t) &= \mathcal{C}_2(t) - \mathcal{C}_2(t-1) + \Delta \mathcal{C}_2(t) \Rightarrow \mathcal{C}_2(t-1)
\end{align*}

Fig. 7: \textit{MASOM} (during performance) aligned with the \textit{Possibility Space} specification.

\text{Creative Output} \{\text{MASOM}\}

\begin{align*}
\mathcal{C}(t) & : \text{Possibility space} \\
c(t) & : \text{Current musical object} \sim \text{instance of } \mathcal{C}(t) \\
\mathcal{M}_1 & : \text{Intrinsic Motivation} \{\text{MASOM}_{GR} = \text{Generation (Random audio from SOM node)}\} \\
\mathcal{M}_2 & : \text{Extrinsic Motivation} \{\text{MASOM}_{GL} = \text{Generation (Machine Listening to Audio Input)}\} \\
t & : \text{Time} \\
\triangleleft, \triangleright : \text{Judgement function}
\end{align*}

\begin{align*}
\triangleleft \mathcal{I}(t) \triangleright (\mathcal{M}_1(t), \mathcal{M}_2(t), c(t)) \\
\downarrow
\triangleleft f(\text{MASOM}_T, T_1 \triangleleft \text{MASOM}_T, T_1 \triangleright) \triangleright (\text{MASOM}_{GR}(t), \text{MASOM}_{GL}(t), c(t))
\end{align*}

Fig. 8: The \textit{Creative output} of the \textit{MASOM} function by my analysis of the published description [13].
5 Comparative analysis suggesting a hybrid implementation architecture

The technologies that form the activity in the MASOM architecture, such as Self-Organising Maps and Variable-Order Markov Models, differ significantly in a technical sense compared to the swarm algorithm and dynamic topological spaces that are the driving technologies in Favoleggiatori 2. However, from the above analyses it is evident that, in the improvising performance situation, MASOM and Favoleggiatori 2 appear very similar in some of their components as aligned with my specification for computational creativity. Both implementations have static concept possibility spaces $\{C_1(t) = C_1(t-1)\}$; neither of these architectures search in concept spaces, in both cases the musical conception is fixed in the implementation coding.

<table>
<thead>
<tr>
<th>Specification component</th>
<th>Favoleggiatori 2</th>
<th>MASOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1(t)$ concept space</td>
<td>static</td>
<td>static</td>
</tr>
<tr>
<td>$C_2(t)$ phenomenon space</td>
<td>$C_2(t-1) \cdot f_{C_2}(c(t), C_2(t-1))$</td>
<td>$MASOM_T$ constant at $t \geq 0$ $MASOM_T$ training at $t &lt; 0$</td>
</tr>
<tr>
<td>$W_1(t)$ memory of possibility space $[C_1(t), C_2(t)]$</td>
<td>$\bigcup_{p=1}^{t-1} \left( C(p) \cdot Q_1 \right)$</td>
<td>$MASOM_T$ constant at $t \geq 0$ represented by SOM</td>
</tr>
<tr>
<td>$W_2(t)$ cultural memory</td>
<td>implicit and constant</td>
<td>implicit and constant</td>
</tr>
<tr>
<td>$Q_1$ attenuation of $W_1$</td>
<td>degrade at constant rate</td>
<td>none</td>
</tr>
<tr>
<td>$Q_2$ attenuation of $W_2$</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>$J(t)$ Imagination</td>
<td>$f(\langle W_1(t), S_3, \langle \langle W_1(t), S_3 \rangle \rangle \rangle)$</td>
<td>$f(\langle W_1(t), T_1, \langle \langle W_1(t), T_1 \rangle \rangle \rangle)$</td>
</tr>
<tr>
<td>$\langle \cdot, \cdot \rangle$ Judgement function</td>
<td>$\langle \cdot, \cdot \rangle_{agent}$ agent behaviour rules</td>
<td>$\langle \cdot, \cdot \rangle_{VOMM}$ VOMM $\rightarrow$ random selection from cluster</td>
</tr>
<tr>
<td>$M_1$ Intrinsic Motivation</td>
<td>agent behaviour rules</td>
<td>$MASOM_{GR}$ Random audio from SOM node</td>
</tr>
<tr>
<td>$M_2$ Extrinsic Motivation</td>
<td>feature analysis</td>
<td>$MASOM_{GR}$ feature analysis</td>
</tr>
</tbody>
</table>

Table 1: Comparison chart of specification components in Favoleggiatori 2 and MASOM during performance.
However, the MASOM exploration of music possibility space occurs during training and not during performance — a distinctive difference from Favoleggiatori 2. As shown in figures 3 the phenomenon space in Favoleggiatori 2 is dynamic \( \mathcal{C}_2(t) = \mathcal{C}_2(t-1) \cdot f_{\mathcal{C}_2}(c(t), \mathcal{C}_2(t-1)) \); this means that the played music changes the music phenomenon possibility space in which Favoleggiatori 2 operates. The phenomenon possibility space of MASOM is encoded through the training of the self-organising map (\( \text{MASOM}_T \)) which is run before performance time; during performance time it appears that the SOM is not altered and so the music phenomenon possibility space is constant \( \mathcal{C}_2(t) = \mathcal{C}_2(t-1) \) as shown in Figure 7. Alternatively we could consider the topological spaces in Favoleggiatori 2 to consist of seven dimensions since they are ‘dynamic’ and change over performance time; so performance time would be the seventh dimension. But this would not align with the formal specification which separates the Possibility Space into concept \( \mathcal{C}_1(t) \) and phenomenon \( \mathcal{C}_2(t) \) categories that are taken to have a common time dimension \( t \), where the start of performance time is \( t = 0 \). Pre-performance time is then \( t < 0 \) and is when \( \text{MASOM}_T \) training is calculated.

In Table 1 I have aligned some of the specification components with the implementation components to facilitate further comparisons. The Memory \( \mathcal{W}_1(t) \) of the possibility space stands in some relation to the Possibility Space \( \mathcal{C}(t) \) in both implementations: in Favoleggiatori 2 the possibility space memory representation consists of the three topological spaces whereas in MASOM the possibility space memory is the \( \text{MASOM}_T \) (the trained SOM). Thus there is some overlap, or conflation, between the phenomenon space \( \mathcal{C}_2(t) \) and the memory \( \mathcal{W}_1(t) \); in MASOM the two components are equal \( \mathcal{W}_1(t) = \mathcal{C}_2(t) \). In Favoleggiatori 2 however there is a programmed attenuation \( Q_1 \) of \( \mathcal{W}_1(t) \) over time; in other words memory is gradually degraded at a constant rate during performance time. In [12] I speculate that errors in memory, as compared to the explored possibility space may have a direct impact on the ‘novelty’ in Boden’s sense [1], or the deviation from the normal, of the creative output. In other words the difference between \( \mathcal{C}_2(t) \) and \( \mathcal{W}_1(t) \) may have a positive effect on the novelty of the Creative Output. The implementation of the memory attenuation \( Q_1 \) in Favoleggiatori 2 is an application of this positive concept of memory ‘error’.

As shown in Table 1 the Imagination function \( \mathcal{J}(t) \) of the two implementations are equivalent. Also in Table 1 the Judgement functions \( \ll . \gg \) are represented in the agent behaviour rules in Favoleggiatori 2 (from Figure 4) and in the VOMM and random cluster-member selection process in MASOM (from Figure 8).

Given this analysis I could imagine a hybrid (see Figure 9) of the two architectures taking advantage of their similarities and differences. In such a hybrid the Judgement function \( \ll . \gg_{\text{hybrid}} \) could be a function that employs a combination of both of the two original architectures \( \ll . \gg_{\text{agent}} \) and \( \ll . \gg_{\text{VOMM}} \). For example: using the Favoleggiatori 2 architecture as the framework, the VOMM

---

4 Another recent implementation approach emphasising the potential positive effect of memory error in an improvising machine is Kalonaris’s ‘Dory’ which is ‘purposively flawed and forgetful’ [7].
Computational creativity specification for music: comparative analysis

The \( \mathcal{C} \) function processes the known musical phenomenon space \( \mathcal{E}_2(t) \) also using the current musical event \( c(t) \) as input. The \textit{Imagination} function \( \mathcal{J}(t) \) is entirely based on the memory of the possibility space \( \mathcal{E}_1(t), \mathcal{E}_2(t) \) where \( \mathcal{E}_1(t) \) is constant since the concept space is fixed (rather than dynamic as per the specification) in the encoding of the implementation. The hybrid \textit{Judgement} function with \textit{Imagination} \( \mathcal{J}(t) \) then becomes the union of second order functions where each agent has an individual \textit{Judgement} function using the output of the first order \( \mathcal{J}(t) \) function.

In the hybrid model there remains overlap, or conflation, between \textit{Intrinsic Motivation} \( \mathcal{M}_i \) and the \textit{Judgement} function \( \mathcal{J}(t) \) since both express the agent behaviour rules. However, using the hybrid \textit{Judgement} function \( \mathcal{J}(t) \) as a second order function, results in a

\[
\begin{array}{l}
\mathcal{C}(t) : [\mathcal{E}_1(t), \mathcal{E}_2(t)] \text{ Possibility space (represented topologically)} \\
c(t) : \text{Current musical object – instance of } \mathcal{C}(t) \\
[P(t), R(t)]_a : \text{topological subspace local to agent } a \\
\mathcal{M}_i : \text{Intrinsic Motivation} \{ \text{Favolaggiatori 2 agent behaviour rules} \} \\
\mathcal{M}_e : \text{Extrinsic Motivation} \{ \text{analysis of human performance} \} \\
\implies \mathcal{J}(t) : \text{Judgement function} \\
n : \text{number of agents} \\
\end{array}
\]

\[
\begin{array}{l}
[P(t), R(t)]_a \subset \mathcal{E}_2(t) \\
[P(t), R(t)]_a = \mathcal{J}(t) \Rightarrow \mathcal{VOMM} (c(t), \mathcal{M}_i(t)) \\
\mathcal{J}(t) = f(\mathcal{M}_i(t)) \\
\mathcal{J}(t) \Rightarrow \mathcal{hybrid} = \bigcup_{n=1}^{n} \left( \mathcal{J}(t) \Rightarrow \mathcal{VOMM} (c(t), \mathcal{M}_i(t)) \Rightarrow \mathcal{agent}(a) \right) \\
\end{array}
\]

Fig. 9: A hybrid Creative Output using components from MASOM in the Favolaggiatori 2 architecture.
differentiation between the hybrid $M_1$ and $J$. In relation to the specification
this differentiation would appear to be an improvement, as identity between com-
ponents will tend to make the specification structure implode, in the sense that
the identity-related components of the specification become indistinguishable.

To the components of Favolaggiatori 2 the hybrid model adds the VOMM
responsiveness to the current musical situation $c(t)$, probably increasing the
adaptability of the resulting machine improviser. To the components of the MA-
SOM architecture the hybrid model adds a dynamic memory space and a swarm
based judgement function to replace the MASOM randomised choices, possibly
giving a more linear musical responsiveness in the hybrid improviser. An imple-
mation of the hybrid model is needed to test these aspects in performance
situations with human performers and such an implementation will be the next
step for this direction of research.

6 Conclusion

The specification expresses a formal structuralist approach to computational
creativity using components developed heuristically from conceptualisation of
human creativity. However, I take human creativity and computational creativity
as two distinct logical categories, and no identity is implied between these
two categories (as I have argued in more detail in [11] and [12]). The creators
of MASOM were certainly not aiming to accord with my specification for com-
putational creativity. On the other hand, in my implementation I have been at
least partly motivated by my development of the specification. However, both
implementations can be aligned with the specification as shown, and this kind
of analytical alignment has been productive in terms of suggesting a hybrid ap-
proach for a new improvisation machine. I propose this analytical work as a
measure of success for the specification as a tool for investigating and represent-
ing computational creativity.

I aim to give the specification the widest possible scope as a formal definition
of computational creativity. However, comparisons between implementations
aligned with the specification are not intended to evaluate these implementa-
tions in terms of being ‘more creative’ or ‘less creative’ because such evaluations
would be misleading, as I have also previously argued [12]. Instead, alignment of
an implementation with the formal specification does allow an evaluation of the
presence or absence of specification components in the implementation as has
been done in sections 3 through 5. Given this, an implementation can be said to
be more or less complete in terms of the specification functionalities. However, a
‘degree of completeness’ of functionality is not necessarily equivalent to a ‘degree
of creativity’ in the output of the system employing that functionality. A partial
creativity which includes some subset of specification components, such as was
shown in the analyses of Favolaggiatori 2 and MASOM, may result in an output
that is deemed ‘creative’.

The specification is prescriptive regarding what computational creativity
could be, but not regarding how it should be implemented; it does not pro-
vide any pseudocode from which implementations could be implied. In other words, the specification states what a computational creativity could include, but not how to include it. In this sense my current specification development has a different focus from recent theories about computational creativity such as the ‘FACE’ and ‘IDEA’ models from Colton, Charnley and Pease [2] which emphasise particular technologies.

One might draw analogy between individual components in the specification and nodes in a network system; such a system may be a computational creativity and so I expect that varying implementations may include members of computational creativity component families, and the specifics of the implementation of a component may depend on the system context. I expect this will be inclusive, such that for example members of the Imagination component family $\mathcal{I}(t)$ and Judgement function $\ll . \gg$ can include those presented for Favolaggiatori 2 and MASOM as well as others such as for example the ‘Associative Conceptual Imagination (ACI) framework’ from Heath, Dennis and Ventura [6, p. 244]. Further analytical work into a broader repertoire of implementations and architectures is likely to deepen the insights available through use of the specification.

Because of limited space, a number of features of the specification have been only lightly touched upon or have not been addressed here at all, such as for example the intertextual context which has been developed elsewhere [12], and specification development is ongoing. I plan to discuss further development of the specification and present more analyses of implementations in future research outputs. The present and subsequent revisions of the specification can serve to analytically approach improvisation machines, as well as computational creativity for music in a wider context, with increasing detail. Also, the specification can give indications of further potential in computational creativity for the next generations of implementations.
Bibliography

