An A-Life Approach to Machine Learning of Musical Worldviews for Improvisation Systems

Marcelo Gimenes and Eduardo R. Miranda
Interdisciplinary Centre for Computer Music Research
University of Plymouth, UK
(marcelo.gimenes, eduardo.miranda)@plymouth.ac.uk

Abstract — **In this paper we introduce Interactive Musical Environments (iMe), an interactive intelligent music system based on software agents that is capable of learning how to generate music autonomously and in real-time. iMe belongs to a new paradigm of interactive musical systems that we call "ontomemetical musical systems" for which a series of conditions are proposed.**

I.Introduction

Tools and techniques associated with Artificial Life (A-Life), a discipline that studies natural living systems by simulating their biological occurrence on computers, are an interesting paradigm that deals with extremely complex phenomena. Actually, the attempt to mimic biological events on computers is proving to be a viable route for a better theoretical understanding of living organisms [1].

We have adopted an A-Life approach to intelligent systems design in order to develop a system called iMe (Interactive Music Environment) whereby autonomous software agents perceive and are influenced by the music they hear and produce. Whereas most A-Life approaches to implementing computer music systems are chiefly based on algorithms inspired by biological development and evolution (for example, Genetic Algorithms [2]), iMe is based on cultural development (for example, Imitation Games [3, 4]).

Central to iMe are the notions of musical style and musical worldview. Style, according to a famous definition proposed by Meyer, is "a replication of patterning, whether in human behaviour or in the artefacts produced by human behaviour, that results from a series of choices made within some set of constraints" [5]. Patterning implies the sensitive perception of the world and its categorisation into forms and classes of forms through cognitive activity, "the mental action or process of acquiring knowledge and understanding through thought, experience and the senses" (Oxford Dictionary).

Worldview, according to Park [6], is "the collective interpretation of and response to the natural and cultural environments in which a group of people lives. Their assumptions about those environments and the values derived from those assumptions." Through their worldview people are connected to the world, absorbing and exercising influence, communicating and interacting with it. Hence, a musical worldview is a two-way route that connects individuals with their musical environment.

In our research we want to tackle the issue of how different musical influences can lead to particular musical worldviews. We therefore developed a computer system that simulates environments where software agents interact among themselves as well as with external agents, such as other systems and humans. iMe's general characteristics were inspired in the real world: agents perform musical tasks for which they possess perceptive and cognitive abilities. Generally speaking, agents perceive and are influenced by music. This influence is transmitted to other agents as long as they generate new music that is then perceived by other agents, and so forth.

iMe enables the design and/or observation of chains of musical influence similarly to what happens with human musical apprenticeship. The system addresses the perceptive and cognitive issues involved in musical influence. It is precisely the description of a certain number of musical elements and the balance between them (differences of relative importance) that define a musical style or, as we prefer to call it, a musical worldview: the musical aesthetics of an individual or of a group of like-minded individuals (both, artificial and natural).

iMe is referred to as an ontomemetic computer music system. In Philosophy of Science, ontogenesis refers to the sequence of events involved in the development of an individual organism from its birth to its death. However, our research is concerned with the development of cultural organisms rather than biological organisms. We therefore coined the term "ontomemetic" by replacing the affix "genetic" by the term "memetic". The notion of "meme" was suggested by Dawkins [7] as the cultural equivalent of gene in Biology. Musical ontomemesis therefore refers to the sequence of events involved in the development of the musicality of an individual.

An ontomemetic musical system should foster interaction between entities and, at the same time, allow for the observation of how different paths of development can lead to different musical worldviews. Modelling perception and cognition abilities plays an important role in our system, as we believe that the way in which music is perceived and organized in our memory has direct connections with the music we make and appreciate. The more we get exposed to certain types of elements, the more these elements get meaningful representations in our memory. The result of this exposure and interaction is that our memory is constantly changing, with new elements being added and old elements being forgotten.

Despite the existence of excellent systems that can learn to simulate musical styles [8] or interact with human performers in real-time ([9-11]), none of them address the problem from the ontomemetic point of view, i.e.:

• to model perceptive and cognitive abilities in artificial entities based on their human correlatives

• to foster interaction between these entities as to nurture the emergence of new musical worldviews

• to model interactivity as ways through which reciprocal actions or influences are established

• to provide mechanisms to objectively compare different paths and worldviews in order to assess their impact in the evolution of a musical style.

An ontomemetic musical system should be able to develop its own style. This means that we should not rely on a fixed set of rules that restrain the musical experience to particular styles. Rather, we should create mechanisms through which musical style could eventually emerge from scratch.

In iMe, software entities (or agents) are programmed with identical abilities. Nevertheless, different modes of interactions give rise to different worldviews. The developmental path, that is the order in which the events involved in the development of a worldview takes place, plays a crucial role here. Paths are preserved in order to be reviewed and compared with other developmental paths and worldviews. A fundamental requisite of an ontomemetic system is to provide mechanisms to objectively compare different paths and worldviews in order to assess the impact that different developmental paths might have had in the evolution of a style. This is not trivial to implement.

A. Improvisation

Before we introduce the details of iMe, a short discussion about musical improvisation will help to better contextualise our system. Not surprisingly, improvised music seems to be a preferred field when it comes to the application of interactivity, and many systems have been implemented focusing on controllers and sound synthesis systems designed to be operated during performance. The interest in exploring this area, under the point of view of an ontomemetic musical system relies on the fact that, because of the intrinsic characteristics of improvisation, it is intimately connected with the ways human learning operates. However, not many systems produced for music improvisation to date are able to learn.

According to a traditional definition, musical improvisation is the spontaneous creative process of making music while it is being performed. It is like speaking or having a conversation as opposed to reciting a written text.

As it encompasses musical performance, it is natural to observe that improvisation has a direct connection with performance related issues such as instrument design and technique. Considering the universe of musical elements played by improvisers, it is known that certain musical ideas are more adapted to be played with polyphonic (e.g., piano, guitar) as opposed to monophonic instruments (e.g., saxophone, flute) or with keyboards as opposed to wind instruments, and so forth.

Since instrument design and technique affect the easiness or difficulty of performing certain musical ideas, we deduce that different musical elements must affect the cognition of different players in different ways.

The technical or "performance part" of a musical improvisation is, at the same time, passionate and extremely complex but, although we acknowledge the importance of its role in defining one's musical worldview, our research (and this paper) is focused primarily on how: (i) music is perceived by the sensory organs, (ii) represented in memory and (iii) the resulting cognitive processes relevant to musical creation in general (and more specifically, to improvisation) conveys the emergence and development of musical worldviews.

Regarding specifically the creative issue, it is important to remember that improvisation, at least in its most generalised form, follows a protocol that consists of developing musical ideas "on top" of pre-existing schemes. In general, these include a musical theme that comprises, among other elements, melody and harmonic structure. Therefore, in this particular case, which happens to be the most common, one does not need to create specific strategies for each individual improvisational session but rather follow the generally accepted protocol.

Despite of the fact that this may give the impression to be limiting the system, preventing the use of more complex compositional strategies, one of the major interests of research into music improvisation relies on the fact that once a musical idea has been played, one cannot erase it. Therefore, each individual idea is an "imposition" in itself that requires completion that leads to other ideas, which themselves require completion, and so on. Newly played elements complete and re-signify previous ones in such ways that the improviser's musical worldview is revealed. In this continuous process two concurrent and different plans play inter-dependent roles: a pathway (the "lead sheet") to which the generated ideas have to adapt and the "flow of musical ideas" that is particular to each individual at each given moment and that imply (once more) their musical worldview.

The general concepts introduced so far are all an integral part of iMe and will be further clarified as we introduce the system.

II. The iMe System

iMe was conceived to be a platform in which software agents perform music related tasks that convey musical influence and emerge their particular styles. Tasks such as read, listen, perform, compose and improvise have already been implemented; a number of others are planned for the future. In a multi-agent environment one can design different developmental paths by controlling how and when different agents interact; a hypothetical example is shown in Fig. 1.

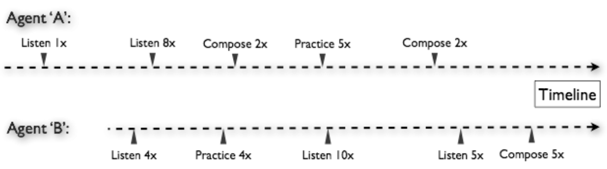


Fig. 1. The developmental paths of two agents.

In the previous figure we see the representation of a hypothetical timeline during which two agents (Agent 'A' and Agent 'B') perform a number of tasks. Initially, Agent 'A' would listen to one piece of music previously present in the environment. After that, Agent 'B' would listen to 4 pieces of music and so forth until one of them, Agent 'A' would start to compose its own pieces. From this moment Agent 'B' would listen to the pieces composed by Agent 'A' until Agent 'B' itself would start to compose and then Agent 'A' would interact with Agent 'B's music as well.

In general, software agents should normally act autonomously and decide if and when to interact. Nevertheless, in the current implementation of iMe we decided to constrain their skills in order to have a better control over the development of their musical styles:

agents can choose which music they interact with but not how many times or when they interact.

When agents perform composition or improvisation tasks, new pieces are delivered to the environment and can be used for further interactions. On the other hand, by performing tasks such as read or listen to music, agents only receive influence.

Interaction can be established not only amongst the agents themselves, but also between agents and human musicians. The main outcome of these interactions is the emergence and development of the agents' musical styles as well as the musical style of the environment as a whole.

The current implementation of iMe's perceptive algorithms was specially designed to take into account a genre of music texture (homophonic) in which one voice (the melody) is distinguishable from the accompanying harmony. In the case of the piano for instance, the player would be using the left hand to play a series of chords while the right hand would be playing the melodic line. iMe addresses this genre of music but also accepts music that could be considered a subset of it; e.g., a series of chords, a single melody or any combination of the two. Any music that fits into these categories should generate an optimal response by the system. However, we are also experimenting with other types of polyphonic music with a view on widening the scope of the system.

In a very basic scenario, simulations can be designed by simply specifying:

• A number of agents

• A number of tasks for each agent

• Some initial music material for the interactions

iMe generates a series of consecutive numbers that correspond to an abstract time control (cycle). Once the system is started, each cycle number is sent to the agents, which then execute the tasks that were scheduled to that particular cycle.

As a general rule, when an agent chooses a piece of music to read (in the form of a MIDI file) or is connected to another agent to listen to its music, it receives a data stream which is initially decomposed into several feature streams, and then segmented as described in the next section.

A. *System's Perception and Memory*

iMe's perception and memory mechanisms are greatly inspired by the work of Snyder [12] on musical memories. According to Snyder, "the organisation of memory and the limits of our ability to remember have a profound effect on how we perceive patterns of events and boundaries in time. Memory influences how we decide when groups of events end and other groups of events begin, and how these events are related. It also allows us to comprehend time sequences of events in their totality, and to have expectations about what will happen next. Thus, in music that has communication as its goal, the structure of the music must take into consideration the structure of memory - even if we want to work against that structure".

iMe's agents initially "hear" music and subsequently use a number of filters to extract independent but interconnected streams of data, such as melodic direction, melodic inter-onset intervals, and so on. This results in a feature data stream that is used for the purposes of segmentation, storage (memory) and style definition (Fig. 2).

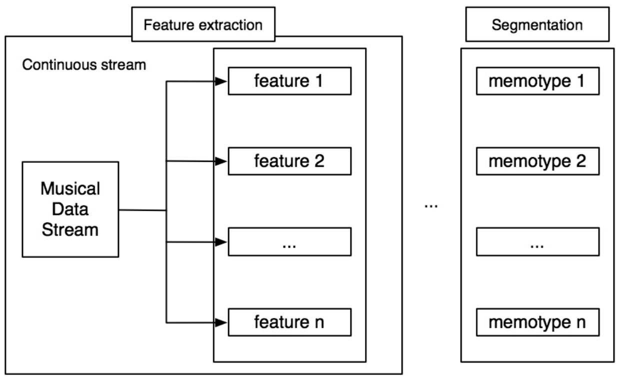
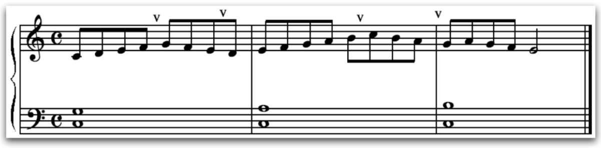


Fig. 2. Feature extraction and segmentation.

To date we have implemented ten filters, which extract information from melodic (direction, leap, inter-onset interval, duration and intensity) and non-melodic notes (vertical number of notes, note intervals from the melody, inter-onset interval, duration and intensity). As it might be expected, the higher the number of filters, the more accurate is the representation of the music. In order to help clarify these concepts, in Fig. 3 we present a simple example and give the corresponding feature data streams that would have been extracted by an agent, using the ten filters:



	1	2	3	4	5	6	7	8	9	10	11	...
a)	0	1	1	1	1	-1	-1	-1	1	1	1	...
b)	0	2	2	1	2	2	1	2	2	1	2	...
c)	120	120	120	120	120	120	120	120	120	120	120	...
d)	120	120	120	120	120	120	120	120	120	120	120	...
e)	6	6	6	6	6	6	6	6	6	6	6	...
f)	2	0	0	0	0	0	0	0	2	0	0	...
g)	5, 7	-2	-2	-2	-2	-2	-2	-2	7, 9	-2	-2	...
h)	120	-2	-2	-2	-2	-2	-2	-2	120	-2	-2	...
i)	960	-2	-2	-2	-2	-2	-2	-2	960	-2	-2	...
j)	6	-2	-2	-2	-2	-2	-2	-2	6	-2	-2	...

Fig. 3. Feature streams, where a) melody direction, b) melody leap, c) melody interonset interval, d) melody duration, e) melody intensity, f) non melody number of notes, g) non melody note intervals from melody, h) non melody interonset interval, i) non melody duration, j) non melody intensity.

Number -2 represents the absence of data in a particular stream. Melody direction can value -1, 0 and 1, meaning descending, lack of and ascending movement, respectively. Melody leaps and intervals are shown in half steps. In streams that hold time information (interonset intervals and duration) the value 240 (time resolution) is assigned to quarter notes. Intensity is represented by the MIDI range (0 to 127); in Fig. 3 this was simplified by dividing this value by ten.

After the extraction of the feature data stream, the next step is the segmentation of the music. A fair amount of research has been conducted on this subject by a number of scholars. In general, the issue of music segmentation remains unsolved to a great extent due to its complexity. One of the paradigms that substantiate segmentation systems has been settled by Gestalt psychologists who argued that perception is driven from the whole to the parts by the application of concepts that involve simplicity and uniformity in organising perceptual information [13].

Proximity, closure, similarity and good continuation are some of these concepts.

Fig. 4 shows a possible segment from piece by J. S. Bach (First Invention for Two Voices) according to Gestalt theory. In this case the same time length separates all except for the first and the last notes, which are disconnected from the previous and the following notes by rests. This implies the application of similarity and proximity rules.

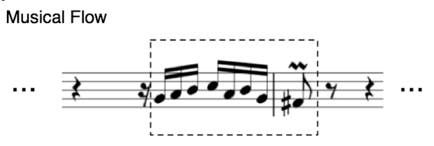


Fig. 4. An example of a music segment.

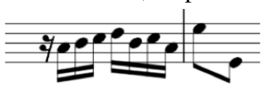
In the example discussed below we decided to build the segmentation algorithm on top of only one of the principles that guide group organization: the occurrence of surprise. As the agents perceive the continuous musical stream by the various expert sensors (filters), wherever there is a break in the continuity of the behaviour of one (or a combination of some) of the feature streams, this is an indication of positions for a possible segmentation. The whole musical stream is segmented at these positions. If discontinuities happen in more than one feature at the same time, this indicates the existence of different levels of structural organization within the musical piece; this conflict must be resolved (this will be clarified later).

In the example of Fig. 3, we shall only consider the melody direction stream ('a' of Fig. 3). Hence, every time the direction of the melody is about to change, a new grouping starts. These places are indicated on the musical score shown in Fig. 3 with the symbol 'v'.

To designate these segmented musical structures we adopted the expression “musical meme” or simply “meme”, a term that has been introduced by Dawkins [7] to describe basic units of cultural transmission in the same way that genes, in biology, are units of genetic information. “Examples of memes are tunes, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperm and eggs, so memes propagate in the meme pool by leaping from brain to brain via a process which, in a broad sense, can be called imitation.” [7].

The idea of employing this concept is attractive because it covers both the concept of structural elements and processes of cultural development, which fits well with the purpose of our research.

A meme is generally defined as a short musical structure, but it is difficult to ascertain what is the minimal acceptable size for a meme. In iMe, memes are generally small structures in the time dimension and they can have any number of simultaneous notes. Fig. 5 shows a meme (from the same piece of the segment shown in Fig. 4) and its memotype representation following the application of three filters: melodic direction, leap and duration:



Mel. direction:	0	1	1	1	-1	1	-1	1	-1
Mel. leap:	0	2	2	1	3	2	4	7	12
Mel. duration:	0	60	60	60	60	60	60	120	120

Fig. 5. Meme and corresponding memotype representation.

Since the memes were previously separated into streams of data, they can be represented as a group of memotypes, each corresponding to a particular musical feature. A meme is therefore represented by 'n' memotypes, in which 'n' is the number of streams of data representing musical features. In any meme the number of elements of all the memotypes is the same and corresponds to the number of vertical structures. By “vertical structure” we mean all music elements that happen at the same time.

B. Memory

The execution of any of the musical tasks requires the perception and segmentation of the musical flow and the adaptation of the memory. As a result, the agents need to store this information in their memory by comparing it with the elements that were previously perceived. This is a continuous process that constantly changes the state of the memory of the agents.

In iMe, the memory of the agents comprises a Short Term Memory (STM) and a Long Term Memory (LTM). The STM consists of the last x memes (x is defined “a priori” by the user) that were most recently brought to the agent's attention, representing the focus of their “awareness”.

A much more complex structure, the LTM is a series of specialized “Feature Tables” (FTs), a place designed to store all the memotypes according to their categories. FTs are formed by “Feature Lines” (FLs) that keep a record of the memotypes, the dates of when the interactions took place (date of first contact - dfc, date of last contact - dlc), the number of contacts (noc), weight (w) and “connection pointers” (cp). In Fig. 6 we present the excerpt of a hypothetical FT (for melody leaps) in which there are 11 FLs. The information between brackets in this Fig. corresponds to the memotype and the numbers after the colon correspond to the connection pointers. This representation will be clarified by the examples given later.

Feature n. 2 (melody leaps):	
Line 0:	[0 0]: 0 0 0 0 0 0 0 0 0 0
Line 1:	[2 2 0 1 0 1 2 5 0]: 1
Line 2:	[1 0 0 3 2 2 0]: 2 20 10 10
Line 3:	[1 0 0 0 1 2 2 4]: 3
Line 4:	[2 0 2 0 4 1 3 0]: 4
Line 5:	[0 3 2 7 0 2 0 4]: 5 8 10
Line 6:	[3 0 2 0 3 2 4]: 6 5 3
Line 7:	[1 0 1 2 2 0 3]: 7 3
Line 8:	[2 0 2 0 2 0 0]: 8 31 8
Line 9:	[2 0]: 47 4 9 9 4 9 9
Line 10:	[5 0 8 2 1 2]: 10

Fig. 6. A Feature Table excerpt.

1) Adaptation

Adaptation is generally accepted as one of the cornerstones of evolutionary theories, Biology and indeed A-Life systems. With respect to cultural evolution, however, the notion of adaptation still seem to generate heated debates amongst memetic theory scholars. Cox [14] asserts that the “memetic hypothesis” is based on the concept that the understanding that someone has on sounds comes from the comparison with the sounds already produced by this person. The process of comparison would involve tacit imitation, or memetic participation that is based on the previous personal experience on the production of the sound.

According to Jan [15] “the evolution of music occurs because of the differential selection and replication of mutant memes within idioms and dialects. Slowly and incrementally, these mutations alter the memetic configuration of the dialect they constitute. Whilst gradualistic, this process eventually leads to fundamental changes in the profile of the dialect and, ultimately, to seismic shifts in the overarching principles of musical organization, the rules, propagated within several dialects."

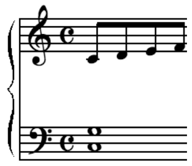
iMe defines that every time agents interact with a piece of music their musical knowledge changes according to the similarities and/or differences that exist between this piece and their own musical "knowledge". At any given time, each memotype for each one of the FTs in an agent's memory is assigned with a weight that represents their relative importance in comparison with the corresponding memotypes in the other memes.

The adaptation mechanism is fairly simple: the weight is increased when a memotype is perceived by an agent. The more an agent listens to a memotype, the more its weight is increased. Conversely, if a memotype is not listened to for some time, its weight is decreased; in other words, the agent begins to forget it.

The forgetting mechanism - an innovation if compared to other systems, such as the ones cited earlier - is central to the idea of an ontomemetic musical system and is responsible for much of the ever-changing dynamics of the weights of memotypes. In addition to this mechanism, we have implemented a "coefficient of permeability" (values between 0 and 1) that modulates the calculation of the memotype weights. This coefficient is defined by a group of other variables (attentiveness, character and emotiveness), the motivation being that some tasks entail more or less transformation to the agent's memory depending on the required level of attentiveness (e.g., a reading task requires less attention than an improvisation task). On the other hand, attributes such as character and emotiveness can also influence the level of "permeability" of the memory.

When a new meme is received by the memory, if the memotype is not present in the corresponding FT, a new FL is created and added to the corresponding FT. The same applies to all the FTs in the LTM. The other information in the FLs (dates, weight and pointers) is then (re)calculated. This process is exemplified below.

Let us start a hypothetical run in which the memory of an agent is completely empty. As the agent starts perceiving the musical flow (Fig. 3), the agent's "sensory organs" (feature filters) generate a parallel stream of musical features, according to the mechanism described earlier. The first meme (Fig. 7) then arrives at the agent's memory and, as a result, the memory is adapted (Fig. 8).



Feature stream:
mdi: 0, 1, 1, 1
mle: 0, 2, 2, 1
mii: 120, 120, 120, 120
mdu: 120, 120, 120, 120

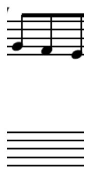
Fig. 7. Meme 1, where mdi is melody direction, mle is melody leap, mii is melody interonset interval and mdu is melody duration.

In order to keep the example simple, we are only showing the representation of four selected features: melody direction (FT1), leap (FT2), interonset interval (FT3) and duration (FT4). Fig. 8 shows the memotypes in each of the Feature Tables. Notice that the connection pointers (cp) of FTs 2 to 4 actually point to the index (i) of the memotype of FT1. The initial weight (w) was set to 1.0 for all of the memotypes and the information date (dfc, dlc) refers to the cycle in which this task is performed during the simulation; in this case, the first task.

i	Memotype	dfc	dlc	noc	w	cp
	Melody direction:					
1	0, 1, 1, 1	1	1	1	1.0	
	Melody leap:					
1	0, 2, 2, 1	1	1	1	1.0	1
	Melody interonset interval:					
1	120, 120, 120, 120	1	1	1	1.0	1
	Melody duration:					
1	120, 120, 120, 120	1	1	1	1.0	1

Fig. 8. Agent's memory after adaptation to meme 1.

Then comes the next meme (Fig. 9), as follows:



Feature stream:
mdi: 1, -1, -1
mle: 2, 2, 1
mii: 120, 120, 120
mdu: 120, 120, 120

Fig. 9. Meme 2.

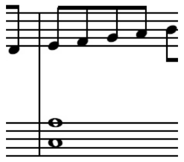
And the memory is adapted accordingly (Fig. 10):

i	Memotype	Dfc	dlc	noc	w	cp
	Melody direction:					
1	0, 1, 1, 1	1	1	1	1.0	2
2	1, -1, -1	1	1	1	1.0	
	Melody leap:					
1	0, 2, 2, 1	1	1	1	1.0	1
2	2, 2, 1	1	1	1	1.0	2
	Melody interonset interval:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	1	1.0	2
	Melody duration:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	1	1.0	2

Fig. 10. Agent's memory after adaptation to meme 2.

Here all the new memotypes are different from the previous ones and stored in separate FLs in the corresponding FTs. Now the memotype of index 1 in FT1 points (cp) to the index 2. Differently from the other FTs, this information represents the fact that memotype of index 2 comes after the memotype of index 1. This shows how iMe keeps track of the sequence of memes to which the agents are exposed. The cp of the other FTs still point to the index in FT1 that connect the elements of the meme to which the memory is being adapted. The weights of the new memes are set to 1.0 as previously.

The same process is repeated with the arrival of meme 3 (Figs. 11 and 12) and meme 4 (Figs. 13 and 14).



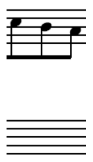
Feature stream:
mdi: -1, 1, 1, 1, 1, 1
mle: 2, 2, 1, 2, 2, 2
mii: 120, 120, 120, 120, 120, 120
mdu: 120, 120, 120, 120, 120, 120

Fig. 11. Meme 3.

i	Memotype	dfc	dlc	Noc	W	Cp
	Melody direction:					
1	0, 1, 1, 1	1	1	1	1.0	2
2	1, -1, -1	1	1	1	1.0	3
3	-1, 1, 1, 1, 1, 1	1	1	1	1.0	
	Melody leap:					
1	0, 2, 2, 1	1	1	1	1.0	1

2	2, 2, 1	1	1	1	1.0	2
3	2, 2, 1, 2, 2, 2	1	1	1	1.0	3
	Melody interonset interval:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	1	1.0	2
3	120, 120, 120, 120, 120, 120	1	1	1	1.0	3
	Melody duration:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	1	1.0	2
3	120, 120, 120, 120, 120, 120	1	1	1	1.0	3

Fig. 12. Agent's memory after adaptation to Meme 3.



Feature stream:
mdi: 1, -1, -1
mle: 1, 1, 2
mii: 120, 120, 120
mdu: 120, 120, 120

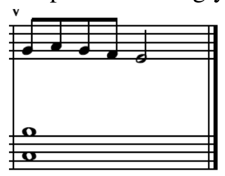
Fig. 13. Meme 4.

The novelty here is that the memotypes for melody direction, interonset interval and duration had already been stored in the memory. Only the melody leap has new information and, as a result a new FL was added to FT2 and not to the other FTs. The weights of the repeated memotypes were increased by '0.1', which means that the relative weight of this information increased if compared to the other memotypes. We can say thereafter that the weights ultimately represent the relative importance of all the memotypes in relation to each other. The memotype weight is increased by a constant factor (e,g, f = 0.1) every time it is received and decreases by another factor if, at the end of the cycle, it is not "perceived". The later case will not happen in this example because we are considering that the run is being executed entirely in one single cycle.

i	Memotype	dfc	dlc	noc	W	Cp
	Melody direction:					
1	0, 1, 1, 1	1	1	1	1.0	2
2	1, -1, -1	1	1	2	1.1	3
3	-1, 1, 1, 1, 1, 1	1	1	1	1.0	2
	Melody leap:					
1	0, 2, 2, 1	1	1	1	1.0	1
2	2, 2, 1	1	1	1	1.0	2
3	2, 2, 1, 2, 2, 2	1	1	1	1.0	3
4	1, 1, 2	1	1	1	1.0	2
	Melody interonset interval:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	2	1.1	2, 2
3	120, 120, 120, 120, 120, 120	1	1	1	1.0	3
	Melody duration:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	2	1.1	2, 2
3	120, 120, 120, 120, 120, 120	1	1	1	1.0	3

Fig. 14. Agent's memory after adaptation to meme 4.

Finally, the memory receives the last meme (Fig. 15) and is adapted accordingly (Figs. 15 and16).



Feature stream:
mdi: -1, 1, -1, -1, -1
mle: 2, 2, 2, 2, 1
mii: 120, 120, 120, 120, 120
mdu: 120, 120, 120, 120, 480

Fig. 15. Meme 5.

i	memotype	dfc	dlc	noc	w	cp
	Melody direction:					
1	0, 1, 1, 1	1	1	1	1.0	2
2	1, -1, -1	1	1	2	1.1	3, 4
3	-1, 1, 1, 1, 1, 1	1	1	1	1.0	2
4	-1, 1, -1, -1, -1	1	1	1	1.0	
	Melody leap:					
1	0, 2, 2, 1	1	1	1	1.0	1
2	2, 2, 1	1	1	1	1.0	2
3	2, 2, 1, 2, 2, 2	1	1	1	1.0	3
4	1, 1, 2	1	1	1	1.0	2
5	2, 2, 2, 2, 1	1	1	1	1.0	4
	Melody interonset interval:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	2	1.1	2, 2
3	120, 120, 120, 120, 120, 120	1	1	1	1.0	3
4	120, 120, 120, 120, 120	1	1	1	1.0	4
	Melody duration:					
1	120, 120, 120, 120	1	1	1	1.0	1
2	120, 120, 120	1	1	2	1.1	2, 2
3	120, 120, 120, 120, 120, 120	1	1	1	1.0	3
4	120, 120, 120, 120, 480	1	1	1	1.0	4

Fig. 16. Agent's memory after adaptation to meme 5.

C. Generative Processes

Gabora [16] explains that, in the same way that information patterns evolve through biological processes, mental representation - or memes - evolves through the adaptive exploration and transformation of an informational space through variation, selection and transmission. Our minds perform tasks on its replication through an aptitude landscape that reflects internal movements and a worldview that is continuously being updated through the renovation of memes.

In iMe agents are also able to compose through processes of re-synthesis of the different memes from their worldview. Obviously, the selection of the memes that will be used in a new composition implies that the musical worldview of this agent is also re-adapted by reinforcing the weights of the memes that are chosen.

In addition to compositions (non real-time), agents also execute two types of real-time generative tasks: solo and collective improvisations. The algorithm is described below.

1) Solo improvisations

During solo improvisations, only one agent play at a time, following the steps below

a) Step 1: Generate a new meme according to the current "meme generation mode"

The very first memotype of a new piece of music is chosen from the first Feature Table (FT1), which guides de generation of the whole sequence of memes, in a Markov-like chain. Let us assume that the user configured FT1 to represent melody direction. Hence, this memotype could be, hypothetically [0, 1, 1, -1], where 0 represents "repeat the previous note", 1 represents upward motion and -1 represents downward motion. Once the memotype from FT1 is chosen (based on the distribution of probability of the weights of the memotypes in that table), the algorithm looks at the other memotypes at the other FTs to which the memotype at FT1 points at and chooses a memotype for each FT of the LTM according to the distribution of probability of the weights at each FT. At this point we would end up with a new meme (a series of n memotypes, where n = number of FTs in the LTM).

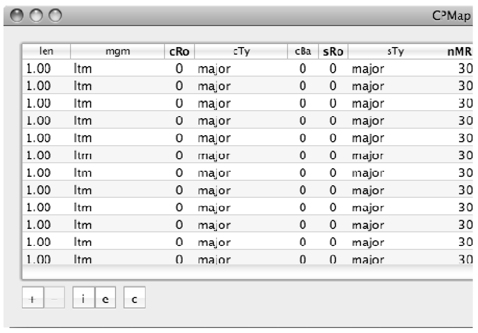
The algorithm of the previous paragraph describes one of the generation modes that we have implemented: the "LTM generation mode". There are other modes. For instance, there is the "STM generation mode", where agents choose from the memes stored in their Short Term Memory. Every time a new meme is generated, the agent checks the Compositional and Performance Map (explanation below) to see which generation mode is applicable at any given time.

b) Step 2: Adapt the memory with the newly generated meme

Once the new meme is generated, the memory is immediately adapted to reflect this choice, according to the criteria explained in the previous section.

c) Step 3: Adapt the meme to the Compositional and Performance Map (CPM)

The new meme is then adapted according to criteria foreseen at the CPM. The CPM (Fig. 17), iMe's equivalent to a "lead sheet", possesses instructions regarding a number of parameters that address both aspects of the improvisation: the generation of new musical ideas and the performance of these ideas. Examples of the former are: the meme generation mode, transformations to the meme, local scales and chords, note ranges for right and left hand. Examples of the latter are: ratio of loudness between melodic and non-melodic notes, shifts for note onset, loudness and duration both for melodic and non-melodic notes. Instructions regarding the performance only affect the sound that is generated by the audio output of the system and is not stored with the composition.



len	mgm	cRo	cTy	cBa	sRo	sTy	nMR
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30
1.00	ltm	0	major	0	0	major	30

Fig. 17. A CPM excerpt.

The instructions (or "constraints") contained in the CPM are distributed on a timeline. The agent checks the constraints that are applicable at the "compositional pointer", a variable that controls the position of the composition on the timeline, and acts accordingly.

d) Step 4: Generate notes and play the meme (if in real time mode)

Until this moment, the memes are not real notes but only meta-representations described by the memotypes (melody direction, melody leap, etc.). Given the previously generated notes and the CPM, the "actual notes" of the meme must be calculated and sent to a playing buffer.

e) Step 5: Store the meme in the composition

An array with the information of the sequence of the memes is kept with the composition for future reference and tracking of the origin of each meme. There is another generation mode, the "MemeArray generation mode", where an agent can retrieve any previously generated meme and choose it again during the composition.

f) Step 6: Repeat previous steps until the end of the CPM

The agent continuously plays the notes of the playing buffer. When the number of notes in this buffer is equal to or less than 'x' (parameter configured by the user), the algorithm goes back to step 1 above and a new meme is generated until the whole CPM is completed.

2) Collective improvisations

The steps for collective improvisations are very similar to the steps for solo improvisations, except for the fact that the agents play along with a human being. We have implemented this task as two separate sub-tasks (a listening sub-task and a solo improvisation sub-task) running in separate threads. Memes are generated as in a solo improvisation and the agents' memory is equally affected by the memes they choose as well as by the memes that they listen from the musical data originated by the external improviser. Both agent and external improviser follow the same CPM.

At the end of the improvisation (solo or interactive), the composition is stored in the system in order to be used in further runs of the system.

III. CONCLUSIONS AND FURTHER WORK

In this paper we introduced Interactive Musical Environments (iMe) for the investigation of the emergence and evolution of musical styles in environments inhabited by artificial agents, under the perspective of human perception and cognition. This system belongs to a new paradigm of interactive musical systems that we refer to as "ontomemetical musical systems" for which we propose a series of prerequisites and applications.

As seen from some of the experiments that we have presented, we understand that iMe has the potential to be extremely helpful in areas such as the musicological investigation of musical styles and influences. Besides the study of the development of musical styles in artificial worlds, we are also conducting experiments with human subjects in order to assess iMe's effectiveness to evaluate musical influences in inter-human interaction. The study of creativity and interactive music in artificial and real worlds could also benefit with a number of iMe's features, which we are currently evaluating as well.

The memory of an agent is complex and dynamic, comprising of all memotypes, their weights and connection pointers. The execution of musical tasks affects the memory state in proportion to the appearance of different memes and memotypes. A particular musical ontomemesis can thereafter be objectively associated with the development of any agent's "musicality".

Bearing in mind that iMe can be regarded as a tool for the investigation of musical ontomemesis as much as a tool for different sorts of musicological analyses, a series of different simulation designs could be described.

Future improvements to the system will include the introduction of algorithms that would allow iMe to become a self-sustained artificial musical environment such as criteria to control the birth and demise of agents and the automatic definition of their general characteristics such as attentiveness, character, emotiveness, etc. Agents should also possess the ability to decide when and what tasks to perform, besides being able to develop their own Compositional and Performance Maps.

ACKNOWLEDGMENT

The authors would like to thank the funding support from the Brazilian Government's Fundacao Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior (CAPES).

REFERENCES

1. Miranda, E.R., *The artificial life route to the origins of music.* Scientia, 1999. **10**(1): p. 5-33.
2. Biles, J.A. *GenJam: A Genetic Algorithm for Generating Jazz Solos*. in *International Computer Music Conference*. 1994.
3. Miranda, E.R., *Emergent Sound Repertoires in Virtual Societies.* Computer Music Journal, 2002. **26**(2): p. 77-90.
4. Miranda, E.R., *At the Crossroads of Evolutionary Computation and Music: Self-Programming Synthesizers, Swarm Orchestras and the Origins of Melody.* Evolutionary Computation, 2004. **12**(2): p. 137-158.
5. Meyer, L.B., *Style and Music: Theory, History, and Ideology*. 1989, Philadelphia: University of Pennsylvania Press.
6. Park, M.A., *Introducing Anthropology: An Integrated Approach*. 2002: McGraw-Hill Companies.
7. Dawkins, R., *The Selfish Gene*. 1989, Oxford: Oxford University Press.
8. Cope, D., *Computers and Musical Style*. 1991, Oxford: Oxford University Press.
9. Rowe, R., *Interactive Music Systems: Machine Listening and Composing*. 1993: MIT Press.
10. Pachet, F., *Musical Interaction with Style.* Journal of New Music Research, 2003. **32**(3): p. 333-341.
11. Assayag, G., et al. *Omax Brothers: a Dynamic Topology of Agents for Improvization Learning*. in *Workshop on Audio and Music Computing for Multimedia, ACM Multimedia*. 2006. Santa Barbara.
12. Snyder, B., *Music and Memory: An Introduction*. 2000, Cambridge, MA: MIT Press.
13. Eysenck, M.W. and M.T. Keane, *Cognitive Psychology: A Student's Handbook*. 2005: Psychology Press.
14. Cox, A., *The mimetic hypothesis and embodied musical meaning*. MusicæScientiæ, 2001. **2**: p. 195–212.
15. Jan, S., *Replicating sonorities: towards a memetics of music.* Journal of Memetics - Evolutionary Models of Information Transmission, 2000. **4**.
16. Gabora, L., *The Origin and Evolution of Culture and Creativity.* Journal of Memetics, 1997.