INTERACTIVE COMPOSING AS THE EXPRESSION OF AUTONOMOUS MACHINE MOTIVATIONS

Peter Beyls
Interdisciplinary Centre for Computer Music Research
University of Plymouth, United Kingdom &
University College Ghent, Fine Arts Department, Belgium

peter.beyls@plymouth.ac.uk

ABSTRACT

This paper documents a novel model supporting rewarding musical human-machine interaction based on the idea of mutual influence rather than the specification of explicit, scripted interaction protocols. A Biology inspired computational model is suggested containing networks for listening, playing and the unsupervised synthesis of autonomous machine motivations. Motivations are assembled from non-linear relationships that interpret external changes, implemented in the drive object. A drive keeps two competing motivations; (1) integration with a human-suggested context or (2) expression of a native character. A population of musical processing functions is evolved online as to offer musical expertise to fulfil the systems' implicit goal i.e., integration or expression. The shifting musical distance between consecutive statements by human and machine is traced in time as to derive a fitness measure for the musical processing functions currently in use; the service they propose for attaining the goals implied in both basic motivations. Experiments show that man and machine may develop interesting interaction modes without any a priori specifications; the system develops a dynamic personality from the non-linear dynamics emanating from the networked architecture.

1. INTRODUCTION

True improvisation is known to flourish in a social system where inter-personal attitudes, modes of listening and a palette of responsive options develop dynamically as a side effect of the act of improvisation itself. A life-like quality characterizes the explorative nature of musical improvisation and spontaneous play alike. The underlying assumption is that knowledge is developed by continuous activity, by doing something rather than by abstract reflection or conscious planning. This implies taking risks and challenging the unpredictable.

Yet much work in interactive composing (Chadabe 1989) is based on the assumption that genuine improvisation is a linear action-response process, much like an abstract conversation following a primarily fixed set of rules -- a programmer provides musical expertise by a process of one-way specification; rules are formalised in isolation, therefore such systems are not grounded into the act of improvisation itself and consequently, they cannot be considered truly interactive. In addition, rule-based systems have trouble coping with an unpredictable human performer; unanticipated input usually leads to ungraceful degradation in performance. In case one aspires non-idiomatic interaction (Bailey 1980), the behaviour of rule-based systems is considered concentric rather than eccentric; complexity usually follows from a combinatorial explosion of rules though the operational freedom of the human performer is deeply conditioned by the stylistic cranny implied in those rules.

However, various interactive composing programming projects have attempted to circumvent the implicit rigid framework of rule-based systems, let us briefly consider three sophisticated systems. (1) *Cypher* (Rowe 2001) uses the concept of distributed agents: a listener and a player thus consist of many simple software agents (Minsky 1985) that, when properly interconnected, form hierarchical agencies that provide the intended musical expertise. Sophisticated functionality emerges from the interactions amongst the software agent, yet the hierarchies themselves are designed by hand. (2) *Voyager* (Lewis 2000) is a non-hierarchical interactive musical environment consisting of an ensemble of virtual players controlled by global behaviour specifications. It favours a non-hierarchical subject-subject model of musical discourse rather than a stimulus-response model. Voyager makes delicate use of random numbers to favour serendipity and creates decisions based on stochastic analysis of features in a continuous input stream. (3) The primary goal of the *Continuator* (Pachet, 2004) is to have a musical instrument that can learn (using an augmented Markov technique) to play in a specific musical style suggested by a human player. However, the Continuator includes a context-sensing algorithm that turns it in to an interactive musical instrument; information extracted from the most recent context may condition musical material generated from the Markov tables. In conclusion, all three systems suggest forms of explicit, scripted interaction protocols.

In contrast, this paper suggests rewarding interaction to exist as a form of mutual influence. We take inspiration from interaction in biological workspaces (Kugler 1989, Bonabeau and Theraulaz 1997) in suggesting a novel model of human-machine interaction. The hypothesis is that vital features of living systems may guide us to create artificial ones. In short, a living system contains some form of self-representation stretched out in its DNA, a living system exchanges information and energy with a dynamic environment, they rely on the integrity and relationships between their constituent components, they evolve over many generations while learning in-between points of evolutionary breeding. Finally, living systems develop motivations on-line, what to do next does not follow programmed instructions but emerges spontaneously from a set of competing drives.

2. MOTIVATION-DRIVEN INTERACTION

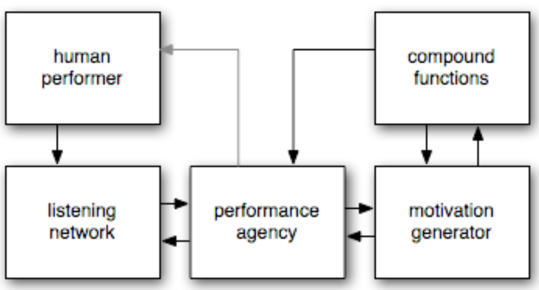

Figure 1: Global systems outline comprising functionality for listening, performance and on-line synthesis of machine motivations

Our system recognises two continuous, conflicting and competing drives: integration and expression. *Integration* means the system is attempting to create musical output matching the current human suggested context and willing to contribute to its further existence. *Expression* implies a system aiming to express a native musical character irrespective of context. The global system consists of three interacting networks (figure 1): (1) an evolved sensor-activator network for the purpose of machine listening (Beyls 2005), a distributed player agency equipped with evolved musical processing functions (Beyls 2007) and finally, networks serving the representation and management of internal machine motivations as introduced here.

Now, the implicit goal is to maximise human-machine *agreement* while in the process of interaction. Agreement takes place when both human and machine strive towards the same orientation (on the average) i.e. both either integration or expression. This implies adaptive behaviour from both parties and the acknowledgement that such interaction objectives cannot be designed by hand.

Motivation driven interaction avoids the need for explicit design specification; a motivation is considered a temporal machine suggestion to either integrate or express. A motivation provides dynamic interpretation of changes in user input, including changes in melodic distance between the last statement produced by human and machine. Motivations contain *relationships* that specify a qualitative, non-linear connection between impinging changes and internal quantities representing the strength of internal motivations.

Four relationships are defined as follows:

R1: if ΔQ-source(t) > 0 then Q-dest(t+1) = Q-dest(t) + ΔQ-source(t)**ƒ*1

R2: if ΔQ-source(t) > 0 then Q-dest(t+1) = Q-dest(t) - ΔQ-source(t)**ƒ*2

R3: if ΔQ-source(t) < 0 then Q-dest(t+1) = Q-dest(t) + ABS(ΔQ-source(t))**ƒ*3

R4: if ΔQ-source(t) < 0 then Q-dest(t+1) = Q-dest(t) – ABS(ΔQ-source(t))**ƒ*4

As an example, consider relation type 1, it implies that a positive change in a source quantity will introduce a positive change in a destination quantity, the amount of change being proportional to the change at the source modulated by the private weighing factor of relation 1. In relation type 2, positive input changes produce negative output changes: output is inverse proportional to input. Note that all relationships { R1 … R4 } feature a private multiplicative weighting factor { *ƒ*1 … *ƒ*4 }.

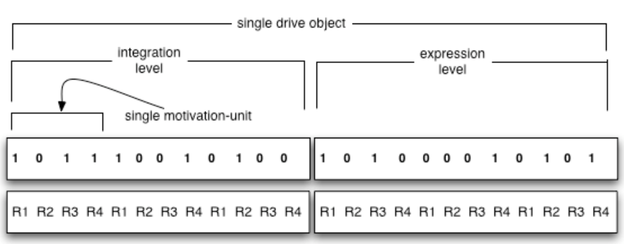

Figure 2: A single drives accommodates changes in three external quantities addressing two basic motivations each articulating a 12-bit relationships vector.

Motivations are implemented as assemblies of competing relationships as seen in figure 2. A single motivation-unit interprets changes in a single external quantity - three such quantities are considered: (1) changes in quantity (density) of user input, (2) changes in the quality (level of interestingness) of user input and (3) changes in similarity between the last sequence produced by the human performer (captured by an adaptive segmentation algorithm) and the last melody played by the system. Every motivation unit is a four bit vector, one bit per relationship { R1 … R4 }, the impact of a relationship is considered when its respective bit is on, it is ignored if the bit is off. When all contributions for a given 12-bit vector are summed, the respective internal level (integration-level and expression-level) is adjusted.

The number of ‘on’ bits in the vector will condition the global responsiveness of the drive. Too many ‘on’ bits may potentially produce over-stimulation leading to erratic output. In contrast, too few ‘on’ bits lead to under-stimulation, in this case, significant changes in input may get lost. We turn to a learning algorithm to discover the proper couplings between input changes and internal motivations. The learning method is similar to reinforcement learning (RL) (Sutton and Barto 1998). RL is a form of unsupervised learning; RL provides only information that the previous action was not appropriate but does not offer instructions of what should be done in order to learn. RL is typically applied in real-world problems characterised by a huge state space. This fits the essence of interactive composing: man and machine must learn to behave successfully without any *a priori* information about their mutual personalities.

The two-stage learning algorithm is depicted in figure 3. Stage one updates the levels of integration and expression from the evaluation of the current relationships. Stage two updates the efficiency according to the current orientation. Stage one encloses two nested loops; the arguments are the current gradients (*delta-values*) in man-machine melodic similarity and the changes in quality and quantity of the contents of working memory. All 24 bits of the *relationships-vector* are addressed (figure 2). The levels are scaled up (activation) or down (inhibition) according to the type of relationship and the delta value. In the end, the integration and expression levels will reflect the accumulated impact of the combination of vector on-bits and the sign of the respective delta-values.

Stage two evaluates the resulting (potentially changed) *drive-orientation*, decided on by taking the highest value of the two competing levels as the winning current orientation. Take note that we exploit *only* the *delta-similarity* at stage two. This delta value and its sign provide information as to whether the current relationships were helpful to steer the drive towards the optimal orientation. The *intended* optimal orientation (integration or expression) is the one that is consistent with the last change in human-machine similarity. For example, when the human-machine melodic distance *decreases* and the orientation is *integration*, we conclude that the drive is indeed resourceful towards the fulfilment of this drive’s orientation – therefore, its efficiency-level is scaled up using the *activation-factor*. Comparably, in case the melodic distance *increases* and the orientation is *expression*, the efficiency level is also scaled up. Efficiency-level is inhibited when the changes in distance are in conflict with the orientation; i.e., either a combination of integration and increasing distance or expression with decreasing distance.

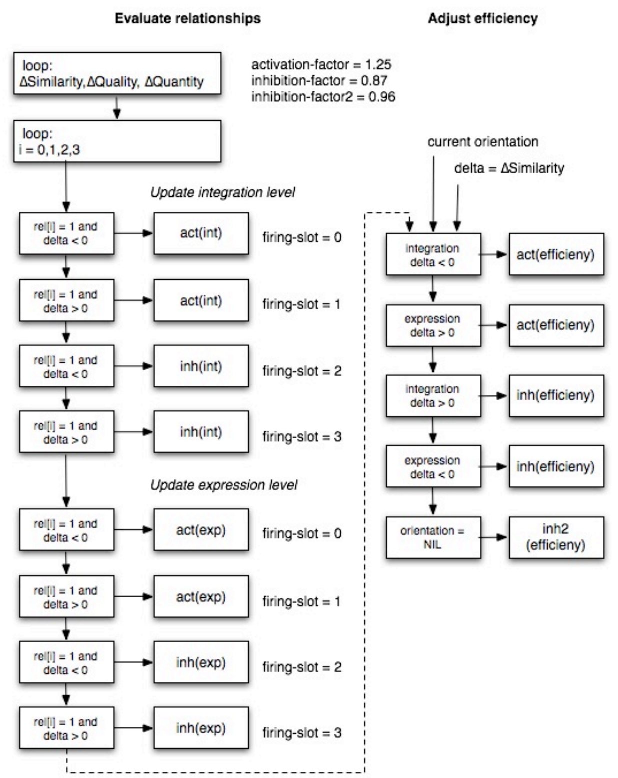

Figure 3: Manipulation of levels of integration and expression according to the evaluation of the drive’s current relationships and adjusting the efficiency of the drive according to perceived changes; i.e., the delta value and its sign.

A given orientation is considered a machine *suggestion* to temporarily approach musical interaction from a given perspective; i.e., either a wish for man and machine drifting apart (expression) or narrowing the man-machine melodic distance (integration). The rationale is that a suggestion is first generated at random and subsequently adjusted according to the evaluation of the data gathered during actual interaction. In total, four situations are considered combining two orientation and two delta signs – the efficiency levels are activated or inhibited according to the motivation as explained above. At last, when there is less than 10% contrast between integration and expression levels, the efficiency of the drive is slightly downscaled using a typical *inhibition-factor2* value of 0.96. In order to optimise of human-machine agreement – the system’s implicit goal – a drive is equipped with one additional instance-variable: its *understanding-level*. This level fluctuates according to the degree of conflict or agreement between the current global orientations of man and machine. To compute this degree, we track and compare two system variables: the machine-global-orientation and the complementary human-global-orientation (scalars between -100 and 100). The listening section (figure 1) adjusts the *human-global-orientation* level according the consecutive changes in similarity between human and machine-produced melodies. It is reasoned that increasing similarity signals a human interactor wishing to integrate, increasing distance indicates individual expression. The amount of contrast between the two options taken over time provides an indication of the human-global-orientation. Similarly, *machine-global-orientation* follows a related algorithm; its level follows the contrast between levels of integration and expression in the current drive.

Both orientations hold a value between 100 (indicative of integration) and -100 (denoting expression), a value around zero thus informs that no clear orientation happens to be available. The degree of *common-understanding* is computed as follows:

```
if (machine-global-orientation.signum == human-global-
orientation.signum
    common-understanding = Agreement
    common-understanding-level =
    (human-global-orientation.abs + machine-global-orientation.abs) / 2
else
    common-understanding = Conflict
    common-understanding-level =
    (human-global-orientation.abs - machine-global-orientation.abs) / 2
```

At this point, the strength and type of human-machine orientation is inferred, *agreement* occurs when both pursue the same orientation, *conflict* when both interactors produce levels of conflicting sign. Finally, the *understanding-level* of the current drive is further adjusted as follows:

```
if common-understanding-level > 0
    ;; activation
    factor = remap(common-understanding-level 0  100 1.0 3.0 )
else
    ;; inhibition
    factor = remap(common-understanding-level 0 -100 1.0 0.3 )

understanding-level = min( 100, understanding-level * factor )
```

The reinforcement factor is proportional to the level of common-understanding and its sign; levels of 0 ~ 100 and -100 ~ 0 are remapped to respectively 1.0 ~ 3.0 and 0.3 ~ 1.0). The *understanding-level* is a learned indication of how successful the drive contributes to an interaction climate characterized by man-machine *agreement*.

The understanding-level (1) and the efficiency (2) of a particular drive are both indications of how well that drive adapts to the current interaction context. The motivation generator (figure 1) contains a population of drives (typically 16) all subject to selection. Typically, the selection process starts with exploration; random selection of drives and evaluation of their performance. This activity gradually articulates a profile of contrasting drives. In a second phase, we aim to exploit the learned expertise; the more promising drives (higher fitness and understanding) become subject to selection. In practice, the selection scheme is driven by an *exploration-exploitation-ratio* – its value being conditioned by the quality and quantity of the most recent human input sequence, further details are documented in (Beyls 2008). We now address the network structure made up of the motivation generator and the compound-function pool and consider their intimately linked functionality.

3. COMPOUND-FUNCTIONS

The higher of the two competing pressures (integration level and expression level) inside a drive will set its final orientation. A drive addresses a population of musical processing functions. These functions keep two records: how well they contribute to integration and to expression. Therefore, a drive can easily select a processing function to fulfil the requirements defined by its current orientation. Once put into action, the processing function will exercise a delayed effect on the efficiency of the instigating drive.

A large collection of simple melodic processing functions was developed, some fairly standard, and some quite original. A variety of basic functions is typically assembled into a compound-function (CF). A CF is instrumental to achieve desired program behaviour during man-machine interaction and can be evolved; the functions and their associated arguments are subject to genetic manipulation using a technique of genetic programming (Koza 1992). Many transformers view the data dimensions (pitch, velocity, duration, inter-onset-time (IOT)) of MIDI events as independent information strata within a single melody.

We developed nine basic functions, including *displacement,* meaning independent rotation of data dimensions, left or right, using individual amounts, timely distortion results without changing any actual values, *scaling* of any dimension (a generalised form of transposition) using a data list of arbitrary length as argument and, finally, *contrast-expansion*. Contrast-expansion is an original transformer in any (combination of) dimension(s), it modifies the dynamic range and the global position of that range. Two parameters condition the effect of this function: the intended bottom and top values. An expansion factor is computed, it will either stretch or compress the current contents of a given dimension – the expressive qualities of a melody are thus amplified or diminished. All basic functions offer simple functionality but when a few are chained into a CF we get rich, often unexpected results that emerge from the interaction of the constituting melodic transformers.

Let us now consider the input to the CF; a principle of multiple influences is maintained. In its basic form, a given CF may process musical material borrowed from the most recent human-produced statement (contents of working memory) or it may address a collection of private data patterns. These patterns constitute a local, idiosyncratic database initially filled with a large number of lists holding random values of variable length. A unique feature of the present system is *the principle of multiple influences*. In short, this principle offers a data selection mechanism to critically merge influences from two input sources defined as *HUMAN* (current human suggested context) and *EGO* (a subset of the current private patterns). The amount of influence from a given source is proportional to how many dimensions (pitch, velocity, duration, IOT) are addressed and how much information is taken from them. In addition, when reading particular data, the start position plays a vital role as to the *degree of connection* to the contents of working melody. That connection is also related to the number of data items read, for instance, when the durations of all events in WM are read, a potentially strong connection may result once that information is processed by the current CF. Eventual synchronisation between the reading start-pointers and the ranges of the data also play a role in degrees of recognisability between input data and CF output.

4. EVOLVING MUSICAL PROCESSING FUNCTIONS

The system keeps a population of CF inside its CF-pool, a typical pool looks as follows:

```
id:        0   1  2 3 4 5  6  7 8 9 10 11 12 13 14 15
fitn-I:    0  48  0 0 0 0 69  0 0 0 48  0  0  0  0  0
fitn-E:    0 177 49 0 0 0 67 49 0 0 49  0  0  0  0  0
nr-evs:    9  13  6 4 5 8  8  9 6 9 15  6  5  5  5  9
nrt-used:  0  11  2 0 0 1  6  1 0 0  1  0  0  0  0  0

Integration-expression-ratios: (0 2 2 0 0 0 1 2 0 0 2 0 0 0 0 0)

Integration   :  1
Expression    :  4
No experience: 11
Epoch-nr      :  0

Integration: [ 6 69 ] [10 48 ] [ 1 48 ]

Expression : [ 1 177 ] [ 6 67 ] [10 49 ] [ 7 49 ] [ 2 49 ]
```

The first row shows the *ID* of every CF. Second and third rows show the current fitness values for the purpose of respectively the integration and expression. Rows four and five document the number of events in the CF and the number of times every CF was applied in the current epoch. Next, the integration-expression-ratios reveal that only a single CF is better for the job of integration (*ID* = 6) and four CF excel for the purpose of expression. When both fitness levels equal zero, the CF is considered to carry no experience. The bottom rows show the individual CF sorted according to fitness for the respective goals. This information is consulted when genetic programming is invoked. Now, let us consider the evaluation of the CF-pool.

A compound-function is considered fit if it contributes effectively towards the realisation of a global systems behaviour that can be characterized as *selfish* – selfish because the fitness of the current CF is adjusted only by considering the current drive, irrespective of information gathered by the listening network (other modes exist, however beyond the scope of this paper).

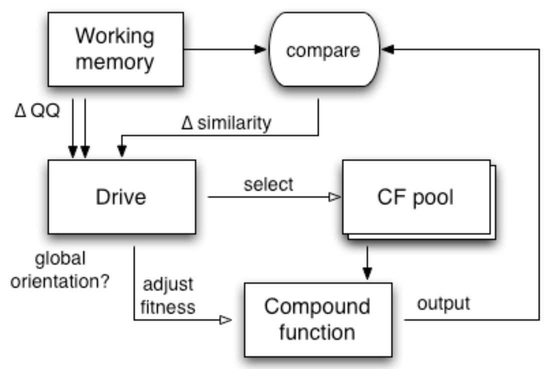

Figure 4: Changes in similarity between the contents of WM and CF are interpreted by the drive while its resulting orientation decides how the fitness of the CF is updated.

The fitness of a given CF is always proportional to how well it manages to contribute to the realisation of an interaction climate implied by the system's current orientation i.e., the *machine-global-orientation* as defined by the current drive. Figure 4 reveals the delicate interaction between CF and drive. The changes in similarity between WM and CF (Δsimilarity) are interpreted by the relationships inside the drive in addition to the changes in quality and quantity of WM (ΔQQ). The drive also selects a new CF from the CF-pool by considering its global orientation, for instance, when it is *integration*, the drive will select a CF that proved to be helpful to assist the process of integration (bringing human and machine musically closer) in previous interactions.

The fitness of a CF is adjusted as detailed next. When an orientation happens to be available (i.e., given enough contrast between levels of integration and expression), three options are considered; (1) when man and machine are getting musically together *(current-similarity > previous-similarity)* and the orientation equals *integration*, the integration-fitness is incremented because this CF was successful in attaining the required functionality. In case the distance increases, the integration-fitness is downscaled. A similar procedure considers the change in distance given an orientation equaling *expression*. Finally, a tolerance of 5% is introduced to judge the delta-similarity (the absolute value of the difference between the current and the previous man-machine similarity). If less than 5%, *both* fitness levels are decremented. The global result is that fitness levels are pushed up and down according to the correlation between current orientation and changes in musical distance. The rationale, then, is to balance the selection of particular compound-functions according to their accumulated fitness and the intended functionality; i.e., either expression or integration. This introduces the archetypal difficulty of biased selection: either select the fittest function obtained so far or, in contrast, select a random function that may prove to develop improved functionality in the future. This typifies the dilemma of *exploration vs. exploitation.* In short, this problem is addressed by having an *exploration-exploitation-ratio*, a quantity that changes in proportion to the level of human-responsiveness.

The level of *human-responsiveness* is proportional to the average changes in quantity, quality and the *current-no-input-gap* value; the time span since the last note-off event input by the human interactor was received. The rationale of the exploration-exploitation-ratio is to establish a relationship between the intensity of human activity and the urgency of deployment of the information contained by that activity. For instance, if human-responsiveness is high, it is concluded that the human performer is momentarily requesting increased attention, consequently the system should pay attention to the most recent context; in other words, the machine performer should exploit the information just entered by the human performer. In contrast, when human-responsiveness is low, it is concluded that the human performer becomes less interested to interact; the current context is no longer very up-to-date and consequently, it becomes less useful. In that case, we may engage in a process of exploration; select a random CF from the CF-pool and speculate that it may induce renewed interest in the human performer.

The exploration/exploitation problem also plays a crucial role in the breeding process supported by the evolutionary algorithms employed here. A method of genetic programming views the list of simple functions and their arguments as genotype susceptive to the standard operators of cross-over and mutation.

Before breeding the next generation of processing functions, the current generation is analysed, one list is created of CF that feature positive integration-fitness, a second list collects CF with positive expression-fitness. These lists are sorted (high to low) according to the magnitude of the respective fitness and the first two CF are considered parents in the breeding process. For breeding to take place, it is imperative that candidates exist for *both* types of fitness. Next, the sum of the lengths of both lists is computed. A simple weighting scheme is developed to decide which pair of parents provides genotype for crossover. For example, given a typical population size of 16 CF, two pairs of parents leaves (16 minus 4) or twelve CF available for crossover and mutation – the parents themselves are not considered for crossover, however, they are modified by the mutation operator. Now, the number of offsprings created from genotype from either group of parents is proportional to the length if the respective list of candidates.

The foundation of the evolutionary approach here is to cultivate fitness levels according to expertise gained during the time span of one epoch of actual man-machine interaction. The process of reproduction merges aspects of the functionality that led to the current ratio of expertise aimed towards integration and expression. This process repeats forever because a single optimal solution does not exist *and* input from the human performer remains totally unpredictable. Therefore, the breeding procedure is a process of perpetual adaptation.

5. EXPERIMENTAL RESULTS

Figures 5 and 6 document the correlation between the behaviour of the compound-function-pool and the drives-pool in experiment 1.

In figure 5 (documenting 20 evolutionary epochs) fitness levels fluctuate in episodes; more or less regularly spaced at the beginning and more irregular towards the end. Levels develop momentum inside single episodes and a generally incremental fitness for both orientations in the long run. The compound-function pool hence manages to optimise all functions collectively; fitness levels maintain relative stability between epochs. Generally speaking, the expression level supersedes the integration level. Logically, we may conclude that it requires more evolved expertise to integrate with an unpredictable input source and less expertise to express a personal character, that is, sound very different than that input source.

Figure 5 also reveals the development of an incremental expression fitness starting around sample 480 and lasting till sample 880 while integration remains exceptionally steady. Nearly 50% of the evolutionary process (about ten epochs of 20 in total) behaves in a remarkably linear fashion. This is evidence that the CF-pool is able to adapt gracefully facing irregular input. The integration fitness level suddenly boosts at sample 840 and sample 880 signals complete breakdown of expression and integration fitness. Evolution is thus a process of gradual modification *and* abrupt changes, evidence that evolution may be considered a complex dynamical system in itself.

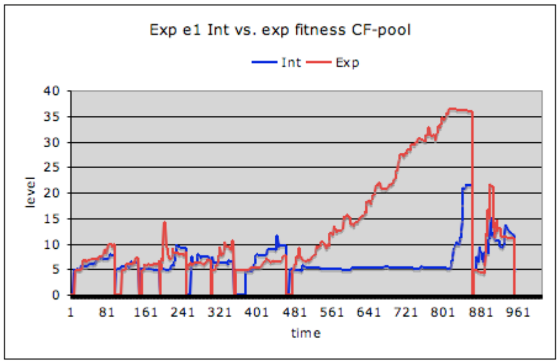

Figure 5: Compound-function pool integration vs. expression fitness levels in experiment 1.

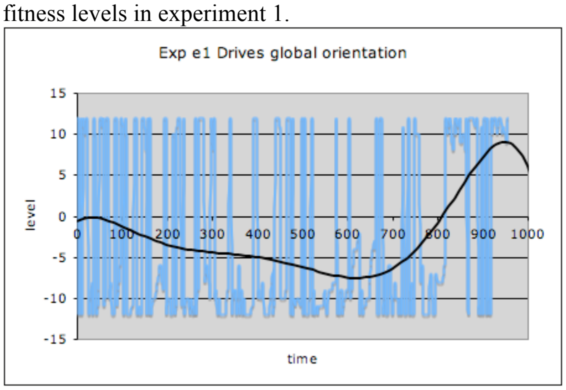

Figure 6: Drives pool global orientation levels in experiment 1.

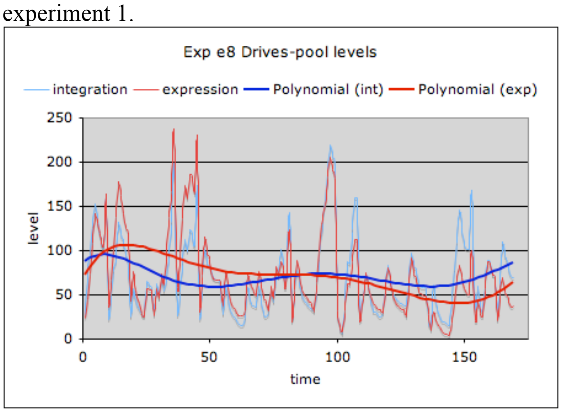

Figure 7: Drives-pool integration and expression levels in experiment 8.

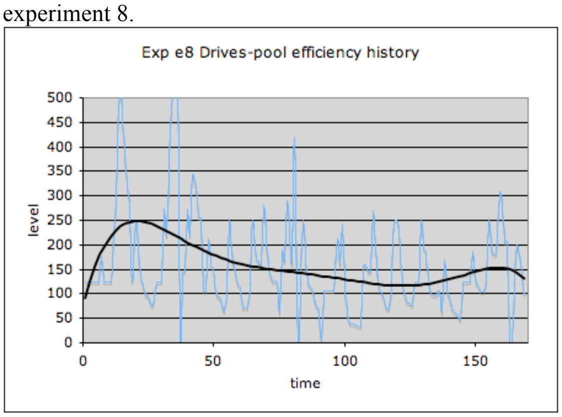

Figure 8: History of drives-pool efficiency in experiment 8.

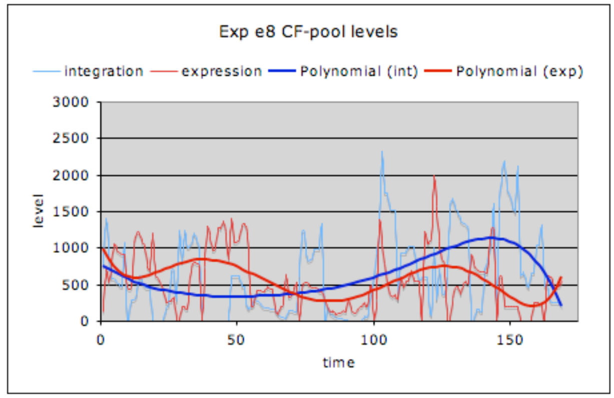

Figure 9: Compound-function pool fitness history for two competing orientations in experiment 8.

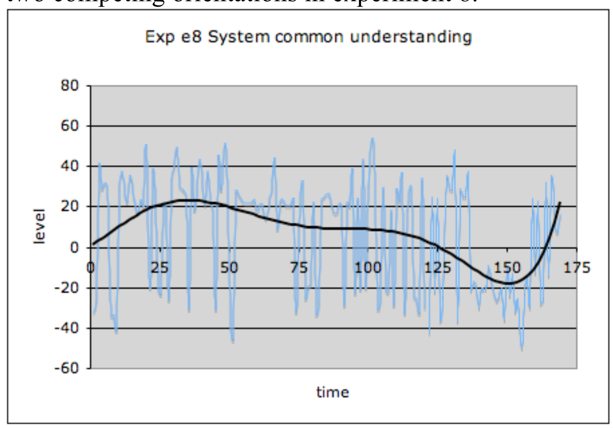

Figure 10: Common-understanding level in experiment 8.

Figure 6 indicates the existence of underlying wave-like behavioural patterns. The global average output values of all current drives in the drives pool are shown. The patterns oscillate between -10 and +10. The signal reveals areas of relative continuity in between sharply peaking swings. The polynomial documents a mainly negative orientation at first; this implies an initial tendency for the machine motivation to prefer *expression*. After sample 800, the orientation becomes *integration*.

The data in figures 5 and 6 demonstrates that CF-pool and drives-pool are correlated in complex ways. Up to about generation 800, pressure for expression (negative values in figure 6) and simultaneous pressure for expression in the CF-pool signal coordinated systems behaviour; an instance of emergent functionality from the interaction between CF-pool and drives-pool. In addition, the breakdown of CF-pool levels at generation 880 coincides with continuous oscillations in the drives-pool – the system is seemingly temporarily disoriented. In the end, the drives-pool settles into a very strong preference (a relative point attractor) for integration while the CF-pool does not develop any significant contrast between integration and expression.

Figures 7 to 10 document behavioural features of experiment 8. Figure 7 shows the evolution of the drives-pool pressures for integration and expression, towards the end, integration is the globally winning orientation. A wavelike behaviour is seen, exemplary of the competing activity between both basic orientations. Both levels also roughly follow each other in between epochs, again revealing a tight coupling between both orientations. Figure 8 shows the drives-pool peaking efficiency levels, in this case, the efficiency polynomial shows a nearly flat data profile. This discloses that the non-linear oscillations inside a drive have a more profound behavioural impact than the otherwise anticipated incremental impact of evolutionary optimisation. Figure 9 shows the very articulated fitness levels for the purpose of respectively integration and expression of the compound-function pool.

A strong correlation is seen between drives levels (figure 7) and drives fitness (figure 9); the data in both figures is linked in complex ways, reflected in polynomials that are nearly synchronized. Note that the CF-pool incrementally develops higher contrast between both fitness levels, indicative of a gradually acquired power to discriminate successfully, in this case, a preference for integration.

Figure 10 documents systems behaviour at a high level of abstraction; the common-understanding level (section 2) is depicted. Positive values signify human-machine *agreement*, negative mean *conflict*. The system clearly oscillates though agreement dominates up to generation 125 and a transitory phase exists around generation 150. Again, the global effect of the very large number of non-linear connections spread out in the various networks comprising the system manages to develop non-trivial well-characterised global behaviour. The occurrence of the transitory phase may well be associated with the increased potential instability entailed by the relative inability of the system to develop expertise for the purpose of expression prominent at generation 150 in figure 9.

5. CONCLUSION

This paper shows interesting human-machine interaction to emerge from the non-linear interactions propagating a networked computational model. It takes inspiration from emergent behaviour as seen in biological workspaces. We suggest learning and evolution as an alternative to explicit, scripted interaction protocols. In addition, the fitness bottleneck typically challenging selectionist methods is avoided by introducing a implicit fitness measure; changes in melodic distance between human and machine generated material is interpreted in relation to the systems' current orientation. The motivation generator documented here successfully coordinates the interplay of machine suggestions (motivations formalised as drive objects) and a body of musical expertise (a population of complex processing functions). Experiments provide evidence that the dynamic coupling between the interacting networks offers the intended emergent functionality; man and machine may develop rewarding interaction modes such as mutual agreement as a side effect of interaction itself.

6. REFERENCES

[1] Bailey, D. 1980 *Improvisation: its nature and practice*, Moorland Publishing in association with Incus Records, UK

[2] Beyls, P. 2005 *Evolving Adaptive Sensors in a Synthetic Listener*, Proceedings of the International Computer Music Conference, Barcelona, Spain

[3] Beyls, P. 2007 *Interaction and Self-Organization in a Society of Musical Agents*, Music and Artificial Life Conference, Proceedings of ECAL, Lisbon, Portugal

[4] Beyls, P. 2008 *On-line development of Man-Machine Relationships: Motivation-driven Musical Interaction*, Proceedings of the Generative Arts Conference, Celestino Soddu (Ed.), Milan, Italy

[5] Bonabeau, E. and Theraulaz, G. 1995 *Why Do We Need Artificial Life?* In: Artificial Life: An Overview, C. Langton (Ed.), The MIT Press, Cambridge, MA

[6] Chadabe, J. 1989 *Interactive composing; an overview*, The Music Machine, C. Roads (Ed.), The MIT Press, Cambridge, MA

[7] Koza, J. 1992 *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, MA

[8] Kugler, P. (Ed.) 1989 *Self-organization in Biological Work Spaces*, Elsevier Science Publishers BV, Holland

[9] Lewis, G. 2000 *Too many notes: computers, complexity and culture in Voyager*, Leonardo Music Journal 10: 33-39

[10] Minsky, M. 1985 *The Society of Mind*, Simon and Schuster, New York

[11] Pachet, F. 2004 *Playing with Virtual Musicians: The Continuator in practice*, IEEE Multimedia 9(3): 77-82

[12] Rowe, R. 2001 *Machine Musicianship*, The MIT Press, Cambridge, MA

[13] Sutton, R. and Barto, A. 1998 *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, MA