Generative Improv. & Interactive Music Project (GIIMP)

Ian Whalley
Department of Music
The University of Waikato
Hamilton, New Zealand
+64 7 8562889
musik@waikato.ac.nz

ABSTRACT
GIIMP addresses the criticism that in many interactive music systems the machine simply reacts. Interaction is addressed by extending Winkler's [18] model toward adapting Paine's [10] conversational model of interaction. Realized using commercial tools, GIIMP implements a machine/human generative improvisation system using human gesture input, machine gesture capture, and a gesture mutation module in conjunction with a flocking patch, mapped through microtonal/spectral techniques to sound. The intention is to meld some established and current practices, and combine aspects of symbolic and sub-symbolic approaches, toward musical outcomes.

Keywords
Interaction, gesture, genetic algorithm, flocking, improvisation.

1. INTRODUCTION
Real-time human/machine 'interactive' music systems are well established as a platform for music performance /composition, with early conceptual models being summarised by Rowe [12] and Winkler [17]. In the literature, Paine's [10] theoretical criticism of early and many later 'interactive' system designs, is that the machine simply reacts to rather than interacts with, human agency. Paine [10:297] therefore puts forward a new model of human/machine agency based on the analogy of human conversation. He notes that: the relationship should be unique and personal to participants, unique to the moment of interaction, vary with unfolding dialogue, and be maintained by both parties speaking the same language and addressing the same topic. Moreover, while one party may know where to begin the conversation, and there may be a pre-existing agenda, the terrain of the conversation might be unknown at the outset. Conversation is then about the exchange and sharing of ideas, with participant(s) relationships deepening over time.

Paine goes on to propose that this new model should not be implemented through instrumental pitch and time models, due to their pre-existing musical frameworks, and he suggests (ibid. 301) that composers should not create all the resources needed for a work before its realization, because this inhibits outcomes that might evolve through dynamic interaction. His aesthetic/technical proposition is to use sensing systems that explore input streams rather than individually triggered events.

In addition, he proposes that systems should be realized through object-oriented programming, and be sonically grounded in Wishart's [19] notion of dynamic morphology [see 14 also].

While Paine's conversational model is a useful starting point in advancing interactive music systems research, the limitations have been noted in the literature [see Whalley, 16], as not all conversations are symmetrical in terms of knowledge, participation, and input quality. An interactive system might be used, for example, to educate people about a new style of sound art. The nature of type of conversation may then be asymmetrical, because of the varying knowledge and ability that people might bring to the situation. In addition, there are drawbacks in taking an extreme approach to using new or unknown musical/sonic languages in Paine's model when generating forms and content, because people have to understand, either as audience and/or participant, something of a shared musical/sonic vocabulary/grammar in order to have a conversation. Still, Paine's intended implementation of his model is a useful counterpoint to adopting excessively prescriptive pitch/duration approaches to music/sound art composition. [17].

The most successful attempts toward implementing adaptations of Paine's conversational model are currently founded in generative improvisation approaches to human/machine agency, and particularly through applying intelligent agent technology [17]. While self-contained machine generative systems [5][8] are common in academic computer music practice, more recent generative improvisation techniques [1] [17] allow for real-time improvised human input as well as human *and* machine adaptation (autonomy and learning) to inputs from both machine and human. Recent successful systems [17] that partly implement Paine's notion of an adaptive relationship between human and machine, demonstrate a balanced human/machine listening/dialogue. However, this is done by first prescribing aspects of sonic language as the basis for a dialogue, rather than prescribing aspect a works form and/or content. These systems demonstrate a modified conversational model because machine and human agency evolve outcomes based on an actual interactive process.

A drawback of recent implementations based in generative improvisation methods, is that they are often built as proprietary systems and are sometimes are more technically than musically engaging. Further they usually use 'audience with system' paradigms [4], rather than a 'performer with system' approach that was common in early 'interactive' methods. Consequently, the new techniques and sonic languages are often less accessible to more mainstream computer music practitioners. GIIMP then aimed to attempt to bridge the gap between some new techniques of interactive generative improvisation, and traditional 'performer with system' perspectives, to allow mainstream electronic musical approaches some access to emerging work but through familiar tools.

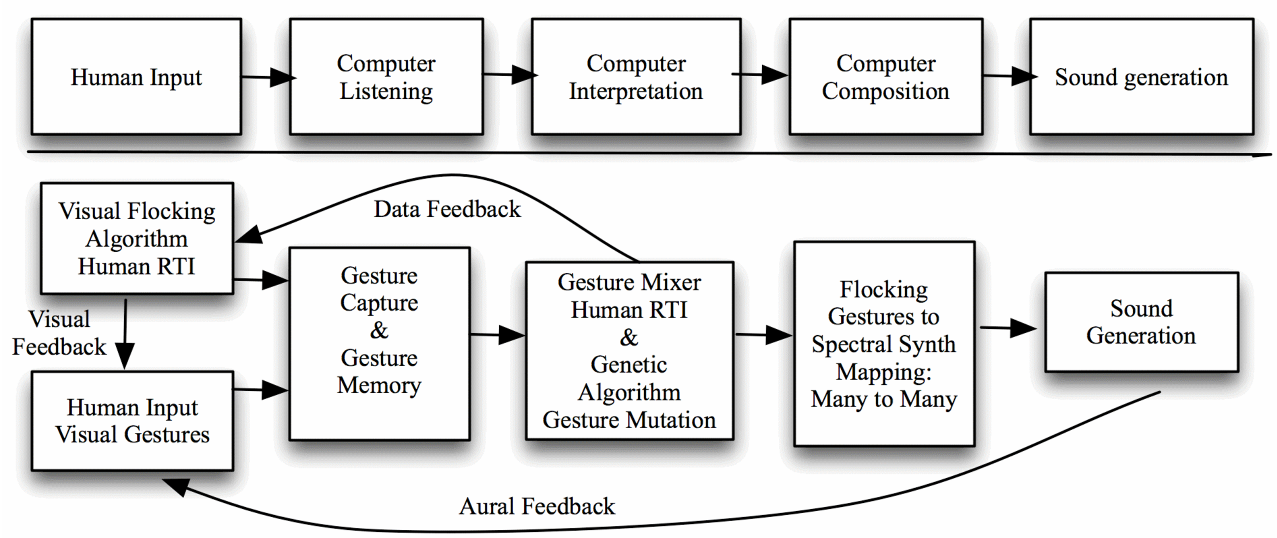

Figure 1. Winkler's model compared to the GIIMP model

2. GIIMP FRAMEWORK

As the 'performer with system' [1] model typified early ideas of 'interaction' when combining computers with traditional instrumental pitch/duration music making methods, it provided a starting point for GIIMP.[2] Winkler's [17] model is given at the top of Figure 1 as a representative approach to build from.

2.1 Linear Model Limitations

Expanding the criticism of the lack of feedback and low level of machine agency in early models, Drummond [2] notes that a further limitation was one of drawing on pitch/duration paradigms and instrumental input, as the MIDI technology used (symbolic), saw sound generation as something fixed and triggered rather than something that might evolve through gesture interaction. However, for many mainstream musicians, MIDI as a language, associated set of controllers for real-time performance and sound triggering, remains standard. The first research question was then one of how to modify Winkler's model to accommodate more experimental generation, adaptive learning, and feedback techniques; but still use familiar technology. The 'fixed' nature of the sound would be addressed later when dealing with mapping and synthesis.

2.2 Aesthetic Approach

The second question was what aesthetic approach to take. The decision made about this was partly pragmatic in an attempt to integrate aspects of traditional and contemporary music/sound making. The decision was also a consequence of the view that creating within and extending known archetypes can more carefully balance the needs of innovation *and* communication [9], [16] in artistic work. Accordingly, rather than viewing sonic language as a new invention that might evolve solely out of the interaction of machine/human agency, the language was set in a microtonal/spectral music [10] paradigm at the outset.

[1] Dummond [2] gives a summary of possible approaches and Graugaard [5] a summary of the range of the field.

[2] The recently released *Oxford Handbook of Computer Music* (2009) covers issue of interaction in Chapters 6,8, 9,12, 18.

This is an adaptation of an aspect of Paine's conversational model, in line with recent complex implementations.

The use of these techniques for sound output provided a means to communicate with a musically educated audience, but also avoided taking an extreme approach to using new musical/sonic languages. This is because the intended audience for the sonic output (concert goers) had to understand something of a shared vocabulary/grammar to engage in the communication. Having said this, the GIIMP system could allow for the evolution of forms and unexpected content to emerge from the dynamic human/machine interaction, despite the prescriptive approach to language. Technically, this method also allowed for the separation of control data from generation, more in line with a MIDI control paradigm than an O-O approach, where sound might be the primary data.

2.3 Generic Tools and Hardware

The third research question was one of 'what tools'. As the system was to be used by mainstream electronic musicians, software and tools had to be readily available so they might be familiar with them. In tandem, the aim was to also draw on techniques that might be assessable for more technically literate electronic musicians. Consequently, the use of Max/MSP/Jitter and MIDI mapping seemed central. In this sense, GIIMP takes a similar approach to recent systems such as the *Kinetic Engine* [6] and *Enactiv* [14] that use Max/MSP to implement adaptations of the conversational model through generative improvisation techniques; although GIIMP's implementation, aesthetic underpinning, and intended use (performer with system) differs from these implementations.

2.4 GGIMP Architecture/Approach

An overview of GIIMP's implementation is given in the lower part of Figure 1, mapped against Winkler's model, to show points of difference. Conceptually and drawing on generative improvisation models [17], the method used here allows Winkler's 'reactive' system to move on a continuum toward Payne's conversational model due to i) input being gesture based away but from instrumental models ii) both human and machine agency having similar gesture input into the system iii) beyond gesture capture, there is a gesture memory that allows for the storage of human input iv) the level of input/mix of input between human machine/ agency in terms of gestures to go forward can be set iv) the parameters of the machine input (flocking algorithm) can be changed in real-time by human input if desired v) the machine and human agency/output is influenced by both machine and human input. Most importantly, feedback loops are created as part of the human/machine dialogue (see Figure 1). These allow i) the machine and human agency to learn, remember and evolve as part of the conversation ii) within the constraints of the language set, the human/machine relationship to be unique/personal to participants and moment of interaction, vary with unfolding dialogue, and be maintained by both parties using the same language (gesture) while addressing the same topic (sound language used).

According to Paine's conversational model and apart from prescriptive language, the limitation of the GIIMP model is that the output (sound) is separated from control data, and is arrived at through event triggering. Again, the prescription does not extend to the form/gestures of the work, so retains Paine's notion of a pre-existing agenda in the model. Hence, the terrain of the conversation is partly unknown in advance; and exchange/sharing ideas allows for the relationship between human/agency to deepen over time.

3. METHODOLOGY USED

From this theoretical/aesthetic framework, the methodology involved i) Programming a set of machine based generative music patches with internal mutating rules that could also learn from real-time external human input ii) Mapping (in software) input/output parameters between human/instrument and machine agency iii) Programming machine generative improvisation/real-time human input to musical outputs iv) System testing so that a participating musician could explore the musical possibilities of improvisation with the system.

4. IMPLEMENATION

The system uses a transformative method of data generation, that is gesture driven [12]. The implementation description following relates to Figure 1. For input and data generation, the control system was run on a MacBook using Ableton Live 8 as the platform interface because of its ubiquity and ease of mapping inputs/outputs. The main implementation tool was Max/MSP/Jitter.

4.1 Input

Machine data generation was based on a flocking algorithm, a method already implemented in systems such as DT1 but implemented here in Max/MSP/Jitter. The advantage of flocking algorithms is that they also provide visual representations of the collective gestures of individual agents. In addition, flocking rules can be altered in real-time directly by performers, and programmed to mutate according to routines set within the algorithm. Visual feedback/outputs could also be quickly mapped to MIDI parameters as required.

A MIDI control pad that recorded finger position on an x/y access, as well as speed and pressure, was used for initial human gesture input, and using two pads could allow for gesture layering. The flocking algorithm could be also be mapped to respond directly to human gestures if required, to allow people to initially quickly establish a visual link between machine flocking gestures and human gesture capture in the flocking algorithm's response. GIIMP then relied on machine and human agency speaking the same gesture language, although using different inputs to capture their respective gestures; and both shared the same visual output in the first instance. Conceptually, the gesture approach used in the system was partly influenced by a gesture/morphology understanding of creating sound mass apart from instrumental approaches, argued for by Smalley [15].

4.2 Gesture Capture and Memory

While machine gestures were encoded in the extensive behavioural rules of the flocking algorithm, human gestures that would also influence the flocking algorithm then needed to be captured and stored. The FTM library extension [3] for Max/MSP provided the means to do this. The length of time for the recording of these gestures could be set by the performer; as could the number of them that might be stored and/or discarded. In this sense, while the machine had a perfect memory, the performer could choose what it was useful for the machine to remember of the human gestures. Conceptually, while the machine listens to/remembers human gesture patterns; the performer, when looking at the flocking visual output the machine creates, would not have such a perfect memory. This made the human/machine dialogue asymmetrical.

4.3 Gesture Mixture and Gesture Mutation

The significant part of the human/machine dialogue took place in the next module (see Figure 1 – Gesture Mixture/Mutation). In the first instance, this happens because of the balance of influence/feedback that comes back to the flocking algorithm. If left to it's own device, the flocking algorithm would follow the system's internal rule base, unless these rules were altered by human intervention. The time period could also be set to return to its rule base if it was neglected by human gesture input, and the system would slowly come to a stop if neglected altogether for period. At the other extreme, the machine flocking algorithm might be set to only follow human gesture input. The balance on the continuum between these two possibilities was left to the performer to set and tweak as part of the conversational process. However, the systems also allowed for the machine agent to become increasingly aggressive in the first instance if neglected, but then slowly lose interest over time.

In conjunction, the human gestures that were stored in the system, either inspired by performance ideas or influenced visually by the flocking algorithm beforehand, could be drawn on to bred new gestures through the use of a genetic algorithm programmed in MAX/MSP. Again, the performer could set the level of influence for the feedback from the genetic algorithm to the flocking algorithm. This ensured that the human/machine conversation resulted from a set of interactions that were unique to each session, that could develop over time, and take on their own direction. Moreover, interactions deepened the relationship between human and machine agency as interactions progressed. The system then created a dynamic/complex triggering method.

4.4 Initial Mapping Out

While accepting that the mapping of gesture to sound is often conceptually contentious, it was made simpler here initially because the visual/tactile gesture system in GIIMP had to relate to sonic gestures to make sense integrated sense. Data from the human/machine dialogue collated in the flocking algorithm was then mapped to sound parameters using Max/MSP and MIDI control data. Linking was first based on agents' placement in time on the x/y access, again drawing Smalley's notions of gesture [13]. Accordingly, the vertical axis was mapped to sound frequency, and the extremes of horizontal axis to pan. Tactile pad pressure was mapped to volume, and related to distance on the flocking patch.

[3] See http://ftm.ircam.fr/index.php/Main_Page.

4.5 Further Mapping and Sound

Such simple mappings would result in bland sonic outcomes if using standard MIDI triggering of static sound patches using 12-tet ET tuning alone (see DT1). However, as the input and output of the systems had to be perceptually related to each other and the sonic outcomes had to phenomenological communicate with the audience intended, techniques harmonic/inharmonic spectra manipulation provided a starting point [11] for more subtle and complex outputs. This required decisions about the physical properties of the sound, dynamics of sound, and the mapping between gestures and sonic qualities.

In line with the initial notion of adapting mainstream approaches and using MIDI, a real-time spectral synthesis approach to sound manipulation alone (see Charles [3]) was negated in favour of using micro-tuning techniques to approximate harmonics as the basic scale pitches for a work – a starting technique used by acoustic composers such as Grisey. Some initial micro-tunings could then be made available, but initiated performers might further explore these to suit their needs. Software for this like LMSO, for example, maps to any device that supports the MIDI Tuning Standard (MTS) for this.

Additive synthesis was used to make the sound, because in the first instance it allowed for the adjustment of spectra so that particular timbres seem harmonic as required by adjusting partials to suit to different tunings and scales (see Sethares' method [13]). Further, using a number of additive synthesizer patches with variations on the starting configuration also allowed for spectral reservoirs to be created and drawn on in works. Finally, using additive synthesis meant that a range of other timbral parameters could be manipulated in real-time to create complex and dynamically changing sounds.

Techniques of extending one to many mappings to improve musical responsiveness by using fuzzy logic modules in Max/MSP are demonstrated by Cádiz & Kendall [2], their work using 2 inputs and 6 outputs with response depending on the rule base and fuzzy decision method implemented (also see [4]), mapped out to granular synthesis. Extending the input/output technique here, mapping aspects of agent movements through MIDI to manipulate real-time additive synthesis parameters allowed for greater responsiveness/complexity of dynamic sonic output. In future, the module here might involve increasingly non-linear parameter manipulation for ongoing sound evolution within the human/machine dialogue.

5. TEST & CONCLUSION

First tests of the system from a performer's perspective showed it capable of creating an evolving dialogue based on musical framework as intended. Initial reactions included the desire to be able to further shape the sound with effects manipulation beyond manipulating synthesizer patches and real-time gesture control, to allow greater control of structure. A test in Ableton Live, adding spectral processing effects plugins that could be triggered using additional sections of the hardware controller interface, allowed for this. This kept the aesthetic intent, while again increasing the complexity of the sound. The real-time control of the parameters of spectral effects units might also be put into the mapping model.

To conclude, by using available tools, the intention was to 'bridge performer with system' 'reactive' approaches with more complex generative improvisation approaches to implement an adapted conversational model by prescribing aspects of language to keep engagement/communicative properties central. In balancing the familiar and unfamiliar, GIIMP then implements a simple interactive system toward engaging audiences through some spectral manipulation techniques [11].

[4] http://imte.ircam.fr/index.php/FuzzyLib

6. ACKNOWLEDGEMENTS

My thanks to Katie Reichmuth for FTM library programming.

REFERENCES

[1] Brown, A. Evolutionary Opportunities for Music Composition. *Proc. of Australasian Computer Music Assoc. Conference*. Melbourne, Australia, 2002, 27-34.

[2] Cádiz, R., Kendall, G. Fuzzy Logic Control Tool Kit: Real-time Fuzzy Control for Max/MSP and Pd. *Proceedings* of the *International Computer. Music* Conference, Miami, Florida, USA, 2005.

[3] Charles, J-F. A Tutorial on Spectral Sound Processing Using Max/MSP and Jitter. *Computer Music Journal* 32(3) 2008, 87-102.

[4] Drummond, J. Understanding Interactive Systems. *Organised Sound* 14(2), 2009, 124-133.

[5] Dorin, A. Generative Processes and the Electronic Arts. *Organised Sound* 6 (1), 2001, 47-53.

[6] Eigenfeldt, A. Emergent Rhythms through Multi-agency in Max/MSP. *Lecture Notes in Computer Science.* Springer Berlin / Heidelberg. Volume 4969, 2009, 368-379.

[7] Graugaard, L. Implicit Relevance Feedback in Interactive Music: Issues, Challenges, and Case Studies. *Proceedings of the 1st International Conference on Information Interaction*, Copenhagen, Denmark. ACM International Conference Proceeding Series; Vol. 176 , 2006, 119-128.

[8] McCormack, J., Eldridge, A., Dorin, A., McIlwain, P. Generative Algorithms for Making Music: Emergence, Evolution, and Ecosystems. *The Oxford Handbook of Computer Music,* 2009, 354-379.

[9] Milicevic, M. Deconstructing Musical Structure. *Organised Sound* 3 (1), 1998, 27-34.

[10] Paine, G. Interactivity, where to from here? *Organised Sound* 7 (3), 2002, 295-304.

[11] Reigle R., Whitehead, P. (Eds). Spectral World Musics: *Proceedings of the Istanbul Spectral Music Conference*, Pan Yayıncılık İstanbul, 2005.

[12] Rowe, R. *Interactive Music Systems: Machine Listening and Composing*, MIT Press (1992).

[13] Sethares, W. *Tuning, Timbre, Spectrum, Scale* . Springer-Verlag, 2005.

[14] Smalley, B. Spectromorphology: explaining sound-shapes. *Organised Sound* 2 (2) 1997, 107-126.

[15] Spasov, M. *Enactiv* http://www.keele.ac.uk/depts/mu/staff/miroslav_software.htm accessed 19th Jan, 2010.

[16] Whalley, I. PIWECS: enhancing human/machine agency in an interactive composition system. *Organised Sound* 9 (2) 2004, 167-174.

[17] Whalley, I. Software agents in music and sound art research/creative work: Current state and a possible direction. *Organised Sound* 14(2), 2009, 156-167.

[18] Winkler, T. *Composing Interactive Music. Techniques and Ideas Using Max.* MIT Press, 1998.

[19] Wishart, T. *On Sonic Art.* Routledge, 1996.