Integrated Interactive Music Performance Environment

Dr. Keith Hamel
School of Music
6361 Memorial Rd
University of British Columbia
Vancouver, B.C. Canada
1 -604-822-6308
hamel@interchange.ubc.ca

ABSTRACT
The author has designed and implemented an interactive music performance environment which allows NoteAbilityPro, a music notation and editing program, to be interfaced with Max/MSP/Jitter or Pd. Extensions to NoteAbilityPro allow control messages and notes with additional performance attributes to be embedded in a score, and sent to a network of computers running Max and/or Pd when the score is performed. Score following using the suivi.score object in Max/MSP can be used to synchronize the live performance with the score performance, thereby aligning all control messages with the live performance.

Keywords
Interactive Computer Music, Interactive Performance, Max, Pd, Score following, Music Notation, NoteAbilityPro

1. BACKGROUND
For many years composers and performers working within the field of interactive computer music have been frustrated by the lack of integration between notation software and the interactive performance software commonly used. Typically, a composer will develop an interactive environment in an application such as Max/MSP [12] or Pd [7] which responds to incoming audio signals from acoustic instruments or is triggered by MIDI messages. The composer will usually develop control mechanisms within these programs so that different kinds of processing and/or effects can take place at different times during the composition. If a score is prepared for live performers, this is done in a separate music notation application, and there is usually no direct connection between the notated score and the interactive environment. This lack of connection between the score and the interactive performance application makes simulation and rehearsal of the complete composition difficult, and makes later editing or modifications to the composition awkward since all of the software components used in the composition have to be altered separately.

The author has developed a comprehensive music notation and editing application called NoteAbilityPro [3] which runs on Macintosh computers. Recently, a number of extensions have been added to this program in order to address some of the issues cited above. While this added functionality does not solve all the challenges related to interactive computer music performance, it does provide composers with a much more integrated environment; one that better facilitates rehearsal and pre-rehearsal simulation of the composition. This software environment, called the Integrated Interactive Music Performance Environment (IIMPE), is best suited to composers who are creating interactive computer music compositions with well-defined instrumental parts rather than those working in loose improvisatory situations, though these performances can also benefit from some of the features of this environment.

2. EXTENSIONS TO NOTEABILITYPRO
At the core of IIMPE are extensions to NoteAbilityPro which allow scores to incorporate important performance data in addition to the graphical representation required for score printing and part extraction. While NoteAbilityPro has traditionally emphasized graphical flexibility rather than performance or playback accuracy, most of the recent extensions added to support IIMPE have substantially enhanced NoteAbilityPro's live performance and networking capabilities.

2.1 Embedding Max/MSP or Pd messages
NoteAbilityPro allows Max/MSP or Pd messages to be embedded in a score. These messages are entered into the score as a text class and are located in the score at specific measure and beat locations. The beat location is indicated above the text box, and the location of messages can be freely dragged around the score or copied and pasted elsewhere in the score or into other scores. The format of embedded messages replicates the format that would be used within Max or Pd (a receive name followed by data of arbitrary length and terminated with a semicolon). During score playback, these messages are sent to a receiving application (such as Max/MSP or to Pd) either as UDP network messages, or as System Exclusive MIDI messages. Up to 16 different IP address & port destinations can be specified in NoteAbilityPro and the tracks on which the messages are placed can be directed to any of these destinations. Is it possible, therefore, to direct different streams of data (such as live video controls) to different computers each running different applications. Since the data size of these control messages is very small, no significant latency issues have been encountered – even when using wireless networks.

In the example below, an excerpt of a NoteAbilityPro score shows a flute part (which is intended to be performed live) along with some of the control messages which are to be sent to another computer running Max/MSP. These messages control everything that is happening in Max/MSP over the course of the composition -- the ways in which the flute is processed, the triggering of sound files, samples, and other audio generators, panning controls, etc.

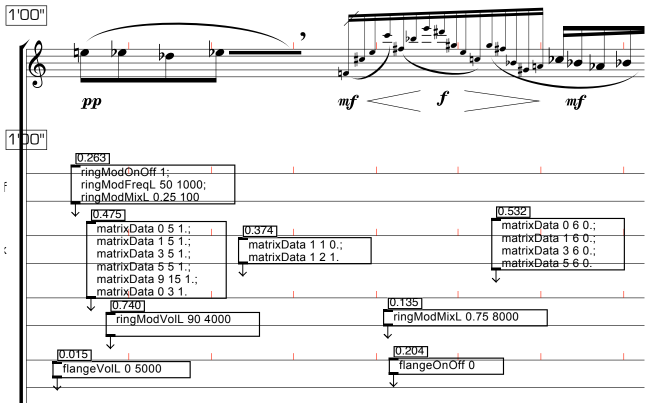

Figure 1. Control messages embedded in NoteAbilityPro

Within Max/MSP or Pd, control messages are received using netreceive [4] objects or with a SysEx parser, each of which pass the message to the corresponding receive object. Some scores may have relatively few embedded messages, while others, such as the example above, may have a single notated instrumental part along with many staves of embedded messages.

2.2 Extended Notes

In order that more complex musical data can also be sent to Max/MSP or Pd, a class of Extended Notes has been added to NoteAbilityPro. Any note entered in the score can be converted to an Extended Note and additional attributes such as microtonal pitch inflection, panning location, and FM harmonicity can be specified. While the default parameters of Extended Notes are mapped to the standard note list data structure used by audio players in the UBC Max/MSP/Jitter Toolbox [3], the data can be interpreted by Max/MSP or Pd in any way the composer desires since they are simply lists of ints and floats. The table below shows the data typically associated with an Extended Note – NoteAbilityPro provides a simple interface for entering and editing Extended Notes.

Table 1. Extended Note Data fields

Data	Value	Data	Value
Pitch	84	C2M Ratio 1	1.215
Velocity	100	C2M Ratio 2	1.0
Channel	1	Stretch Factor	3.5
Pitch Fraction	0.5	Grain Size	150
Duration (ms)	3500	Extra Int 1	0
Pan (LR)	0.28	Extra Int 2	0
Pan (FB)	0.70	Extra Float 1	0.0
Pan (BT)	0.375	Extra Float 2	0.0

2.3 Qlist and Detonate Support

NoteAbilityPro can also save scores or selected parts of scores as Max qlist or detonate objects. These functions allow NoteAbilityPro scores (which might include embedded messages and extended notes) to be embedded in MaxMSP as a qlist (with delta times between events calculated) or to be saved as a detonate object for testing or simulation purposes.

3. CONTROLLING MAX/MSP OR PD

When a score is played by NoteAbilityPro, embedded messages and extended notes are sent to the receiving applications through the designated IP addresses and ports. All tempo changes, repeats and playback settings in the score affect when and how the control messages are transmitted. It is possible, therefore, to simulate a performance of your composition by sending the instrumental outputs from NoteAbilityPro (as audio) into Max/MSP or Pd while having all the control messages and extended notes (which can be generating audio, processing the instrumental sounds, or performing audio analysis) sent as network messages at the same time. Changes to the content of messages in the score or to their placement in the score can be accomplished easily by editing the text objects or by moving the text in the score. In most cases, it is possible to create accurate simulations of the interactive composition without requiring the live instruments to be present. When the composition has been completed, the score will include all notes to played by the performers as well as all the messages and extended notes needed to control all the Max/MSP and/or Pd patches used in the interactive composition.

4. SCORE FOLLOWING

4.1 Manual Following

In the first implementation of the Integrated Interactive Music Performance Environment, score following was done manually by the composer. Score playback was started in NoteAbilityPro, and, with the help of a tempo control panel and the scroll wheel on the mouse, the live performance was loosely synchronized with the instrumental part in the score. With each new performer, a new tempo map was created during rehearsals, and then manual adjustments to the playback tempi were made during performance. While manual score following would not work with all compositions, it proved to be very successful in compositions where some flexibility in the synchronization between the performer and the embedded messages could be tolerated. These interactive performances were much better controlled and much more secure than any previous experiences the author has had with similar performances.

4.2 Automated Score Following

While manual score following suits some compositions, it certainly would not work well with all interactive music, and the fact that it requires a computer operator to be present at all rehearsals and performances is a drawback. The next stage in the development of IIMPE was to implement mechanisms for automated score following. This enables a performer to simply start the NoteAbilityPro score, have their performance tracked using a score follower which in turn adjusts the playback tempo of NoteAbilityPro so that it is aligned with the performer. Once the live performance and the notated score are synchronized, all control messages are automatically aligned to the live performance.

4.3 Score Following using suivi

Various automated score following strategies have been used over the past 20 years. The work of Vercoe [11], Puckette [6], Dannenburg [1], and more recently of Pardo and Birmingham [5], and of Orio and Schwarz [10] provides a wide range of options and approaches, yet the task of achieving accurate and reliable score following in live performance situations remains a significant challenge. While one approach works well for a certain performance situation or music, the same approach may fail miserably in a different performance situation or when following a different kind of score.

In the current implementation of the Integrated Interactive Music Performance Environment, score following using the Max/MSP objects suivi.score and suivi.score~ [8] developed at IRCAM are being tested. While this stage of development is not yet complete, the results so far are very promising. A standard midi file generated by NoteAbilityPro is imported into the suivi.score object and becomes that object's basic internal score representation. Pitch tracking is performed on the live performer (either in Max/MSP or using a Pitch to MIDI converter). The detected pitches are passed to the suivi.score object which sends a synchronization message to NoteAbilityPro through an inter-application or network message. NoteAbilityPro, usually running on a separate computer, receives this message and adjusts its score playback so that it is will be synchronized to a beat location just ahead of the beat location detected by suivi.score.

4.4 Strategies for synchronization

The complexities of score following in live performance situations have been well documented [9], and this environment falls prey to the same problems that all such systems must deal with. Players make mistakes, pitch tracking is sometimes inaccurate, and there are latency issues that must be accommodated. As well, the composer needs to ensure that the entire composition does not break down because an error is made by the performer or a miscalculation is produced by the score following software.

The basic strategy for aligning NoteAbilityPro's playback to the events tracked by the score following software is to make adjustments to NoteAbilityPro's tempo in order that the two scores are more closely synchronized at future events. Currently, there are three modes of synchronization supported by NoteAbilityPro: Fluid, Close, and Tight. Essentially, these three modes control how far in the future the actual synchronization point is. The closer the point of synchronization is in time, the more extreme the tempo changes of the NoteAbilityPro performance are likely to be. When tight synchronization is desired, the score playback tempo is adjusted so that synchronization occurs at the beat location a fraction of a beat ahead of the score following beat location. When loose synchronization is desired, less radical tempo adjustments are made and the synchronization point is further in future. When playing NoteAbility scores that mostly contain control messages, tight synchronization keeps the two score closely aligned. However, when playing scores which include MIDI tracks or other note events, loose synchronization (which result in less extreme tempo adjustments) produces more aesthetically pleasing results.

In order to synchronize the suivi score and the NoteAbilityPro score, a new playback tempo is generated for a short duration, after which the playback tempo is reset to the current tempo. If NoteAbilityPro is running behind the suivi score, the playback tempo is increased, sometimes very dramatically, for a short period of time. If NoteAbilityPro is running ahead of the suivi score, the playback tempo is slowed to allow the suivi score to catch up. The current performance tempo is also updated; it is calculated from the timing of recent synchronization points. The figure below provides a couple of simple examples of tempo adjustments where the synchronization point is 0.5 beats ahead of the beat positions tracked in the suivi score.

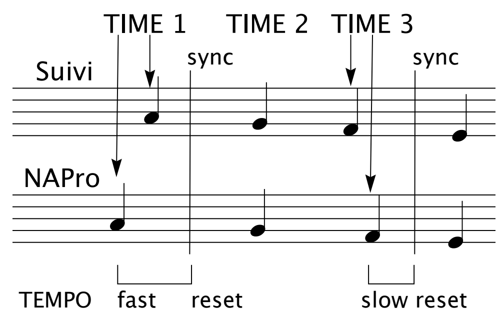

Figure 3. Tempo adjustments with sync point +0.5 beats

4.5 Strengths and Limitations

It might be argued this approach to score synchronization is flawed since the tempo of the playback score is always in flux. There are, however, some practical advantages to this approach, especially for scores having an abundance of control messages rather than timed musical events. First, the fact that the score continues to play (and can never be completely paused or jump to another location in the score) accommodates situations where the suivi gets lost or suggests a score location that it outside a reasonable time horizon – the score will continues to play at the preset tempo until we are back on track or until a cue point in the score is reached. Second, since NoteAbilityPro is able to send messages to Max/MSP at any time, it is easy to programmatically disable score following for musical passages where reliable tracking has proven to be too difficult (e.g. in music containing extended techniques or where heavy noise components are produced by the instruments, or where aleatoric or improvisatory passages are desired). During these passages, the playback score sends a message to the score follower to stop listening and then later to restart score following. The timing and score locations of these control messages can easily be changed during rehearsal.

Currently, the methods used to perform tempo adjustments to the playback score are very simple. We plan to investigate approaches that are able to anticipate tempo trajectories in a performance. We also want to test the applicability of dynamic time warping [8] to the synchronization of the two scores.

5. THE INTEGRATED SCORE

One of the positive results of the developing an interactive computer music composition using this environment is that the score contains a much more complete representation of a composition. Typically, a score created using the Integrated Interactive Music Performance Environment would include the instrumental parts (which can be extracted and given to the players or printed as a complete score and given to the conductor), modified versions of the instrumental parts which are used to generate the suivi scores (since not all the data contained in the complete notated part is necessary for score following), staves containing control messages which will be sent to other applications such as Max/MSP and Pd during performance, as well as tracks containing sound files or standard MIDI data. As well, other kinds of messages (e.g. to control other external devices) may also be included in the score. The more complete the score is, the easier it is for the composer to make changes to the composition (such as inserting or deleting measures, adjusting notes, samples, and control messages, or changing the default tempo map), and the easier it is to manage the data associated with the composition. Finally, having a more integrated score provides a simpler and more complete archive of the composers' creative work.

6. CONCLUSION AND FURTHER WORK

Although this research project is still very much in its infancy, the Integrated Interactive Music Performance Environment has already proven to be a practical and reliable system for designing, testing, simulating, and performing certain types of live interactive computer music. Ultimately, we would like to make this environment as robust and as reliable as possible, and to see it be used in a wide range of musical situations. In order to achieve these goals we will continue to develop and refine the score following system and we will design and test new mechanisms for dealing with score following errors. We also plan to invite a number of composers and composer/improvisers to work with IIMPE in order to assess whether there are stylistic or musical limitations to the system, and to develop strategies for addressing those limitations.

7. ACKNOWLEDGMENTS

I would like to acknowledge the Social Science and Humanities Research Council (SSHRC) of Canada which provided financial support for this research, and the Media and Graphics Interdisciplinary Centre (MAGIC) at UBC which provided research space and administrative support.

Thanks also to Diemo Schwarz, Norbert Schnell, and Serge Lemouton who provide valuable insights and assistance with the suivi objects, and to Bob Pritchard who helped me build the UBC Max/MSP Toolbox modules.

8. REFERENCES

[1] Dannenburg, R. An On-line Algorithm for Real-Time Accompaniment. In *Proceeding of the International Computer Music Conference,* 1984. pp. 193-198

[2] Hamel, K. NoteAbility: A Music Notation System That Combines Musical Intelligence with Graphical Flexibility. In *Proceedings of the International Computer Music Conference*, 1994.

[3] Hamel, K, and Pritchard B. *UBC Max/MSP/Jitter Toolbox.* Retrieved January, 2006 from http://www.opusonemusic.net/muset/toolbox.html.

[4] Mathes, O. netsend and netreceive. Retrieved January 2006 from http://www.akustische-kunst.org/maxmsp/.

[5] Pardo, B. and Birmingham, W. Improved Score Following for Acoustic Performances. In *Proceedings of the International Computer Music Conference,* 2002

[6] Puckette, M. Score Following Using the Sung Voice. In *Proceeding of the International Computer Music Conference,* 1995. pp. 199-200

[7] Puckette, M. Pure Data. In *Proceedings of the International Computer Music Conference*, 1996. pp. 269-272.

[8] Orio, N and Schwarz, D. Alignment of Monophonic and Polyphonic Music to a Score. *Proceedings of the International Computer Music Conference,* 2001.

[9] Orio, N. Lemouton, S. Schwarz, D. and Schnell, N. Score Following: State of the Art and New Developments. In *Proceedings of the International Conference on New Interfaces for Musical Expression* 2003.

[10] Schwarz, D., Orio, N., Schnell, N. Robust Polyphonic Midi Score following with Hidden Markov Models. In *Proceedings of the International Computer Music Conference* 2001.

[11] Vercoe, B. and Puckette, M. 1985. Synthetic Rehearsal: Training the Synthetic Performer. In *Proceedings of the International Computer Music Conference*, 1985. pp. 275-278.

[12] Zicarelli, D., *Max 4.2 Reference Manual,* Cycling '74, www.cycling74.com. 2003.