AN ARTISTIC APPROACH TO DESIGNING VISUALISATIONS TO AID INSTRUMENTAL MUSIC LEARNING

Andrew Johnston
Creativity and Cognition Studios
Faculty of Information Technology
University of Technology Sydney
PO Box 123, Broadway NSW 2007
Australia

Benjamin Marks
36 Longueval St.
Moorooka, Queensland 4105, Australia

Ernest Edmonds
Creativity and Cognition Studios
Faculty of Information Technology
University of Technology Sydney
PO Box 123, Broadway NSW 2007
Australia

ABSTRACT

This paper describes the development of a computer-based music visualisation to support the development of instrumental musical skills in advanced students and professional players. The underlying pedagogical philosophy, based on the "Natural Learning Process" and the emergence of an artistic rather than engineering approach to software development, based on participatory design, are described.

KEYWORDS

Music, learning, teaching, visualization.

1. INTRODUCTION

In this paper we describe the development of software which gives feedback to musicians on a key aspect of instrumental technique – legato (smooth) playing. We first outline past work in the area of computer analysis of live music and discuss our position in respect to instrumental music learning. While the software itself is the main contribution of our work, we feel that the method we used to develop the software – using an artistic approach rather than a more traditional software development methodology – has important implications for those working in this area and may be of interest to others exploring the use of technology in similar domains.

2. PAST WORK

There have been several approaches to using the capabilities of computers to analyse and transform aspects of musical performances to help musicians develop their skills. The approaches range from what might be called the "scientific" approach, where audio sounds are analysed and displayed in the form of graphs to more "artistic" visualisations where the correlation between musical input and computer display is more abstract.

The "Sing-and-See" program (Thorpe 2002), taking the former approach, is designed for singers and singing teachers. The user sings into a microphone and the connected computer displays visualisations of various aspects of the sound. The available displays are a piano keyboard and traditional music notation staff which highlight the note currently being sung, a pitch-time graph showing the pitch of the sung note in comparison to the standard equal-temperament keyboard and a spectrogram and real-time spectrum display showing power in the voice at different frequencies.

Another program which produces similar output - VoceVista (http://www.vocevista.com) - has been used to analyse various aspects of vocal performances (Miller and Schutte 2002). Successfully incorporating these tools into everyday practice requires careful thought, as the complicated nature of the spectrogram display is open to interpretation and may encourage an overly analytical approach.

A project which illustrates a more abstract approach to the mapping between sound and computer response is the "Singing Tree" (Oliver et al. 1997). The singing tree provides both audio and visual feedback to singers in real time. The singer sings a pitch into a microphone and if the pitch is steady the computer provides an accompaniment of consonant harmonies from string, woodwind and vocal chorus. If the pitch deviates, the computer introduces gradually increasing dissonance to the accompaniment including brassier and more percussive instruments and more chaotic rhythms. In addition to the audio feedback, when the singer's note is steady the computer generates video which is designed to give the impression of moving forwards towards an identifiable goal- an opening flower for example. If a steady pitch is not maintained the video reverses. Such an approach emphasises a playful approach and the link between audio input and the audio/visual output is intended to be less deterministic than the more "scientific" mapping between sound and vision used in tools such as "Sing and See".

A prototype tool which analyses performances from another perspective has been created by Nishimoto and Oshima (Nishimoto and Oshima 2001). Their "Music-Aide" program takes standard MIDI (Musical Instrument Digital Interface) input and represents the musical phrases produced by the musician graphically. Different components of the phrases, such as consonant or dissonant notes are displayed in such a way that the relation between these components can be seen.

For example, it might be that when the improvising musician plays phrases containing predominantly firsts and fifths (i.e. 'monochrome chord tones'), she never includes the tritone. Music-Aide would show this characteristic graphically. It is hoped that the musician will therefore be able to notice aspects of their improvising that they might otherwise be unaware of, and thus help them to overcome bad habits or discover new ways of playing.

Another way that Music-Aide might be able to help is in finding new directions and styles. For example, if a musician noticed that the representation of their improvisations always had an empty space in a particular part of the display, they could work out phrases to play that would move the representation into that space. In this way, perhaps tools such as Music-Aide could stimulate the performer to try a style of playing that they had previously not considered.

A tool which displays a view of the relationship between tempo and loudness rather than harmonic aspects is described by Langner and Goebl (Langner and Goebl 2003, Dixon et al. 2002). This tool shows a graph with the x-axis representing tempo and the y-axis the dynamic level. As the music is played, a dot moves around within the graph. As the tempo and loudness vary, the dot moves around the screen, leaving a kind of three-dimensional "trail" behind it that very effectively illustrates the high-level shaping of phrases by the performer. For example, using such a tool the performer might identify patterns in their playing – always slowing down when playing at lower dynamic levels perhaps – that they did not detect aurally.

Johnston, Amitani and Edmonds (2005) describe a 'Virtual Musical Environment' which is designed to encourage a creative approach to improvisation by allowing the performer to control a computer-controlled video screen as well as audio by varying the pitch and volume of the notes they play. The performer moves a cursor around the screen using their instrument and as the cursor lands on certain hot-spots various audio-visual effects are triggered which are intended to encourage exploration of new sounds and links between sound and vision. The effects include recording and playing back short sections of the live performance as well as pre-recorded audio and video.

3. APPROACH TO MUSICAL DEVELOPMENT

Instrumental musicians require skills in two key areas. Firstly they must develop the ability to physically manipulate their body and instrument to produce musical sounds and secondly they require creative skills in order that the music they produce is interesting to others. While various styles of music may emphasise either the craft of instrumental music or the artistic side of musical skill to greater or lesser degrees, musicians generally require a high level of ability in both areas.

3.1 Physical Skill Development

Broadly speaking, pedagogical approaches to the development of the necessary physical skills to play an instrument can be divided into two camps. The first emphasises the need to understand the physiology of instrumental technique in order to facilitate conscious control of the various muscles involved so that "correct" technique can be used. Perhaps due to a desire to approach instrumental music in a more "scientific" way, there is often a tendency for musicians to attempt to take a reductionist approach to improving their playing. For example, singers may attempt to support their sounds by attempting to consciously control their diaphragm in some way in order to improve their range, volume or tone.

Kohut (1992) describes this approach as the "physiological-analysis-conscious-control" method in which musicians attempt to understand the physical actions involved in music making in detail and exert control over them consciously while playing. Kohut argues that this approach is fundamentally flawed and instead prefers the "imitation method" arguing that:

- The muscle manoeuvres involved are too complex and subtle to be meaningfully controlled by the conscious mind and that many muscles (such as the diaphragm) cannot be controlled directly anyway.
- Attempting to consciously control the minutiae of physical actions that take place while playing a musical instrument is at best likely to lead to a tense, mechanical-sounding musical outcome and at worst to "paralysis by analysis" (Frederiksen 1996, Kohut 1992, Stewart 1987), where the musician becomes overwhelmed by the complexity of detailed muscle control and loses the ability to play even simple tunes on their instrument;
- Improved performance results from setting and refining specific musical goals rather than consciously attempting to control physical actions at a low level;

For these reasons a trend in music pedagogy has been towards an approach based on the "natural learning process" (NLP) (Gallwey 1974) which emphasises the importance of leaving the complexities of muscle control to the subconscious so that the conscious mind remains free to set high-level musical goals. The technique for developing new skills is based on imitation, with a strong emphasis on mental musical goal setting and excellence of role models. Teachers taking this approach for example would tend to spend more time on playing for students during lessons and encouraging them to try to copy aspects of the teacher's sound, instrumental technique or musical phrasing and would discourage discussion of physical aspects of instrumental technique.

This position has important implications for those wishing to design software to help instrumental musicians. In particular it would strongly suggest that using computers to analyse the physical actions of the body, rather than the resulting music, is likely to be at best unhelpful and at worst counter-productive.

3.2 Creative Development

Recommendations from the literature for supporting creative development in instrumental music with technology are sparse. However, Nickerson (1999) provides some high-level guidelines aimed at teachers wishing to encourage creativity in their pupils and Shneiderman's (2002) *genex* ("generator of excellence") framework gives some broad but concrete suggestions for those wishing to support creative work with computers. Neither, though, provides us with list of requirements explicit enough to begin software development work. Rather, they may be used as an overarching framework which can guide requirement elicitation during development. Johnston and Edmonds (2004), have taken the guidelines of Nickerson and Shneiderman and extrapolated a set of somewhat more specific but nonetheless high-level guidelines and functional suggestions for those wishing to develop computer-based instrumental music-learning support tools. The work described in this paper was based on these foundations. In particular, the following functional suggestions and goals from this framework were relevant to this project:-

Functional suggestions

- Provide graphical representation of aspects of a performance, such as sound quality, in order that the performer might perceive patterns in, or qualities of, their playing that had previously gone unnoticed;
- Provide support for considering problems (musical and physical) in a different light.

Overarching goals

- Support and encourage development of instrumental technique to facilitate creativity;
- Be wary of 'psuedo-objective' evaluations of sound and other qualitative aspects. Encourage focus on musical end results and avoid triggering 'paralysis by analysis';
- Avoid overly-judgemental feedback. Allow/encourage experimentation in a socially supportive environment;
- Provide structure without stifling innovation and spontaneity.

4. APPROACH TO SOFTWARE DEVELOPMENT

Because it is not possible to extrapolate specific functional requirements for musical visualizations from the literature, we take the view that software development in this case is very much an artistic process. That is, it is not possible to begin with a set of clear, unchanging requirements which are systematically implemented by following a predetermined plan. It was therefore decided that a participatory design approach would be taken, meaning that software development and requirement discovery would take place as part of a collaborative project between software developer (technologist) and musician (artist) (Candy and Edmonds 2002) . The artist in this case was a musician and composer with more than 10 years professional experience. The software developer had a masters degree in computing but in addition had an undergraduate music degree and had worked professionally as a musician. Because of this the artist and technologist were able to communicate freely, which may not have been the case had they both instead had experience exclusively in one domain.

An important realization which was made during the process of development was that while the initial concept had been to design a type of 'tool' for musicians to use in their practice and teaching, a more fruitful approach was instead to concentrate on developing a multimedia performance artwork which was *likely* to also result in software which could also be used in the practice and teaching. Thus the emphasis was on using a creative process to investigate a wide variety of possible musical environments in the hope that an effective work of art responding to musical input might also contain elements suitable to be used as aids for musicians. A parallel might be drawn with etudes composed as pieces of music in their own right which in addition explore and/or expose a particular aspect of instrumental technique. It was therefore decided that the outcome of the creative collaboration would be:

- Software which responded to audio input from the musician by displaying a visual representation of aspects of the music;

- A piece of music which used the software as an extension of the instrument and explored its potential;
- Identification of elements of the software which might be used in practice and/or teaching.

A key aspect of the software development was the use of a ‘visual’ programming environment, Pure Data (pd). Pd is an open-source programming environment which provides powerful audio processing and, with the additional library GEM (Graphics Environment for Multimedia), 3D graphics capabilities. Pd is especially well-suited to collaborative rapid development of audio-visual software for several reasons. In addition to the audio-visual features provided with the language, the fact that the programmer can make changes to the program as it is running is a major advantage in our experience. This allowed the artist to view the output of the program and interact with the software musically while the programmer was making changes. With more generic general-purpose programming languages, such as Java or C++ for example, the program would need to be stopped, recompiled and then restarted for even the most trivial of changes. The process of developing interactive artworks with pd or other similar languages such as Max/MSP has been described as a type of collaborative ‘sketching’ (Weakley et al. 2005) with artist and technologist working together to make rapid adjustments to the software and evaluate their impact quickly. We see this ‘sketchy’ approach as being a very important part of our project.

5. THE WORK

Having spent some time outlining the theoretical underpinnings of this project, we now describe the outcome of the first stage of development. At this stage, we have a performance piece comprising software which visually responds to musical input coupled with a short (approximately 6 minute) composition for solo trombone. The overall piece has been constructed to highlight the absence or presence of smoothness in the music. That is, very smooth legato playing has a clearly different effect on the computer output to either ‘lumpy’ legato playing or non-legato (staccato) playing.

To visually depict smooth playing, it was felt that the display should appear ‘organic’, as if the visualization was a physical sculpture which moved in response to musical input. To achieve this, we use Physical Modelling for Pure Data (pmpd) (Henry 2004), a library for pd which facilitates implementing graphic elements which obey the laws of Netownian physics. Using pmpd, the developer can set up a complete model of a physical system made up of objects joined together by links of given elasticity. In our case, for example, we create a series of spheres which are strung together with elastic links with one of the spheres being fixed in position at the top of the screen (see Figure 1). Once these spheres have been created (and linked to the screen using GEM) it is possible to exert force on individual spheres in any direction and pmpd calculates the effect on the rest of the spheres and links. This means, for example, that if the sphere on the bottom of the chain is “pushed” with a given force for a short period, pmpd can track the position of all the links and spheres and the results can be displayed in real-time on the screen. Thus, the chain of spheres on the screen exhibits very ‘life-like’ behaviour, appearing to respond in the same way an actual physical chain of spheres might in the real world. To simulate gravity, a constant downward force is applied to all spheres. This means that when no sound is detected the spheres hang down vertically (Figure 1).

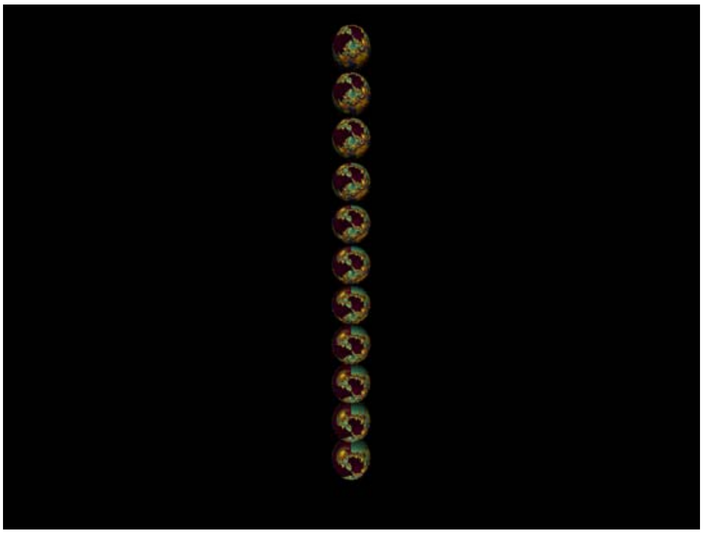

Figure 1. Visual output from the software when the performer is silent.

In order that the display can respond to musical input, it is necessary to link the spheres to the performer's microphone. The pd object fiddle~ (Puckette et al. 1998) is used to identify several aspects of the sound including the pitch and volume of the note being played as well as the pitch and volume of its component harmonics. The mapping used is simple, with each individual pitch-class or note (C, C#, D, etc.) being mapped to one of the spheres. If a note is not in tune, the software simply "rounds" it up or down to the nearest pitch. The volume of the note determines the amount of force applied to the sphere. This means that if a steady pitch is played, the sphere associated with that particular note moves out to one side and appears to float as if held by an invisible hand (see Figure 2).

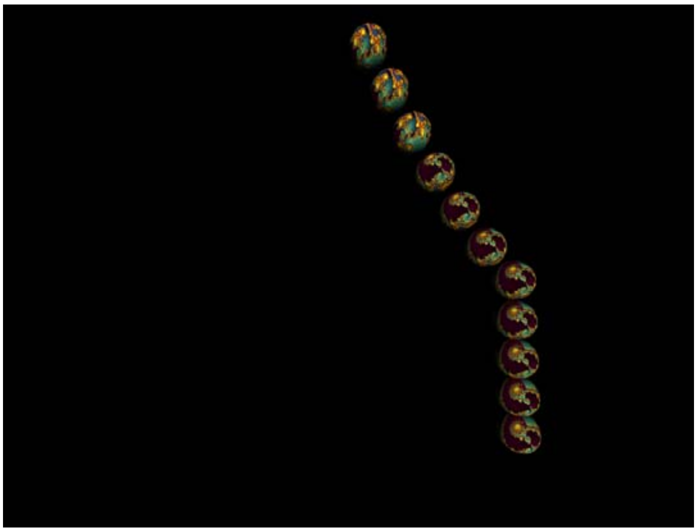

Figure 2. Visual output when the performer holds a steady pitch. It can be seen that the note linked to the 5th sphere from the bottom is currently sounding. The varying brightness of the spheres gives feedback on the harmonic structure of the sound.

Apart from the main note played by the performer, fiddle~ identifies harmonic components of the sound and these are mapped to the rotation and brightness of the spheres. The volume of the harmonic determines both the brightness of the sphere and how far it 'twists' from its default position. This means that if the performer holds a steady pitch and changes the timbre of the note they see the sphere associated with that pitch float out to one side and also see other spheres in the chain changing brightness and spinning in response to the changing timbre.

6. REFLECTION

The software in its current state is quite effective in highlighting the difference between legato (smooth) and staccato (detached) playing, at least on the trombone. For example, if many different short detached notes are played the sphere chain dances around in many directions. If, on the other hand, the performer slurs smoothly between two steady tones the sphere linked to the first pitch will float out to one side and then the sphere linked to the second tone will gently take its place. If the slur is not well executed, the chain of spheres will bump and jostle around at the change of notes. We feel that this characteristic of the software is potentially useful in both instrumental music practice and in teaching as a means of improving musicians' and students' awareness the characteristics of their legato playing. For example, it is often the case that students do not detect a problem with their legato playing because they are distracted by the physical requirements of their instrument. Using this software, a teacher could play a legato phrase and have the student observe the results and then compare the response when they play the phrase. In this way the software could provide a kind of short-cut, helping the student become aware of aspects of their playing that they were previously unaware of. Of course it would be foolish to claim that this software could be a substitute for a good teacher or that this tool will be used continuously during lessons and practice, and these are not our aims. We believe however, that it has potential to be a useful additional tool for performers and teachers to use in specific situations.

An additional outcome of this project has been the development and exploration of an approach to software development in this domain. Through our review of the literature and our work developing this software we have come to realize that:

- Requirements for tools to support the development of creative skills in fields such as instrumental music are ill-defined. The rapid advancement of technology has made sophisticated signal processing and 3D graphics feasible on consumer-level computers, but how (or if) to apply this technology usefully in this area is an area for further research.
- A fruitful way to explore this area and potentially discover useful applications is to pair software developers with musicians and composers to explore the artistic potential of the technology. The collaborative development of multimedia "etudes" such as the one described here give both artists and technologists freedom to experiment in a way that is more consistent with the artistic nature of musicians than a more traditional software engineering approach.

7. CONCLUSION AND FUTURE DIRECTIONS

We have described a multimedia software "etude" which was developed as a composition for live performance which explored a particular aspect of instrumental technique – legato playing. The theoretical foundations of our approach to musical skill learning and software development have been outlined and the resulting software and its behavior discussed.

Further work remains to be done. Thorough evaluation of the software in a practice/learning context is necessary in order to determine how it might be put to practical use and to help discover how effective or ineffective our "artistic" software development method is. Of course, the method of evaluation must be chosen carefully, as it is possible that musicians and teachers may wish to use the software in interesting ways which we have not considered. However, it is important that the next phase of development be informed by careful evaluation of the current software and considered reflection upon the validity of our theoretical approach.

REFERENCES

Candy, L. and Edmonds, E. 2002, *Explorations in Art and Technology*, Springer-Verlag, London.

Dixon, S., Goebl, W. and Widmer, G. 2002, "Real Time Tracking and Visualisation of Musical Expression", *Music and Artificial Intelligence: Second International Conference,* vol. 2445 / 2002, (eds) Anagnostopoulou, C., Ferrand, M. and Smaill, A., Springer-Verlag Heidelberg, Edinburgh, Scotland, UK, pp. 58-68.

Frederiksen, B. 1996, *Arnold Jacobs: Song and Wind*, Windsong Press Limited, Gurnee, Illinois.

Gallwey, T. 1974, *The Inner Game of Tennis*, Random House, New York.

Henry, C. 2004, "Physical Modeling for Pure Data (Pmpd) and Real Time Interaction with an Audio Synthesis", *Sound and Music Computing '04,* Paris.

Johnston, A., Amitani, S. and Edmonds, E. 2005, "Amplifying Reflective Thinking in Musical Performance", *Creativity and Cognition 2005,* London.

Johnston, A. and Edmonds, E. 2004, "Creativity, Music and Computers: Guidelines for Computer-Based Instrumental Music Support Tools", *Australasian Conference of Information Systems*, (eds) Elliot, S., Williams, M.-A., Williams, S. and Pollard, C., School of Information Systems, University of Tasmania, Hobart, Tasmania.

Kohut, D. L. 1992, *Musical Performance: Learning Theory and Pedagogy,* Stipes Publishing L.L.C, Champaign, Illinois.

Langner, J. and Goebl, W. 2003, "Visualising Expressive Performance in Tempo-Loudness Space", *Computer Music Journal*, vol. 27, no. 4**,** pp. 69-83.

Miller, D. G. and Schutte, H. K. 2002, *Documentation of the Elite Singing Voice* [online]. Available: http://www.vocevista.com/contents.html [Accessed June 12, 2003].

Nickerson, R. S. 1999, "Enhancing Creativity", in *Handbook of Creativity*, ed. Sternberg, R. J., Cambridge University Press, Cambridge, pp. 392-430.

Nishimoto, K. and Oshima, C. 2001, "Computer Facilitated Creating in Musical Performance", *Scuola Superiore G. Reiss Tomoli*, L'Aquila, Italy.

Oliver, W., Yu, J. and Metois, E. 1997, "The Singing Tree: Design of an Interactive Musical Interface", *Proc. Designing Interactive Systems,* ACM Press New York, NY, USA, Amsterdam, The Netherlands, pp. 261-264.

Puckette, M. S., Apel, T. and Zicarelli, D. D. 1998, "Real-Time Audio Analysis Tools for Pd and Msp", *International Computer Music Conference,* International Computer Music Association, San Francisco, pp. 109-112.

Shneiderman, B. 2002, *Leonardo's Laptop*, MIT Press, Cambridge, Massachussetts.

Stewart, M. D. (ed.) 1987, *Arnold Jacobs: The Legacy of a Master,* The Instrumental Publishing Company, Northfield, Illinois.

Thorpe, W. 2002, *Visual Feedback of Acoustic Voice Features in Singing Training* [online]. Available: http://www.med.rug.nl/pas/Conf_contrib/Thorpe/Thorpe_bio_touch.htm [Accessed June 20, 2003].

Weakley, A., Johnston, A., Edmonds, E. and Turner, G. 2005, "Creative Collaboration: Communication, Translation and Generation in the Development of a Computer-Based Artwork", *11th International Conference on Human-Computer Interaction,* Las Vegas, Nevada.