‘Spheres of Influence’ - An Interactive Musical Work

Andrew Johnston
Creativity and Cognition Studios
University of Technology Sydney
PO Box 123, Broadway, NSW 2007, Australia
+61 2 9514 4497
aj@it.uts.edu.au

Benjamin Marks
36 Longueval St.
Moorooka, Queensland 4105, Australia
+61 7 3848 5875
ysoltandben@optusnet.com.au

Ernest Edmonds
Creativity and Cognition Studios
University of Technology Sydney
PO Box 123, Broadway, NSW 2007, Australia
+61 2 9514 4640
ernest@ernestedmonds.com

ABSTRACT
In this paper we describe the development of an interactive artwork which incorporates both a musical composition and software which provides a visual and aural accompaniment. The system uses physical modeling to implement a type of virtual ‘sonic sculpture’ which responds to musical input in a way which appears naturalistic. This work forms part of a larger project to use art to explore the potential of computers to develop interactive tools which support the development of creative musical skills.

Categories and Subject Descriptors
J.5.3 [**Arts and Humanities**]: Performing Arts.

General Terms
Performance, Experimentation, Human Factors, Languages.

Keywords
Music, interaction, music-learning, artistic exploration.

1. INTRODUCTION
In this paper we describe the development of a computer-based artwork which comprises a musical composition for trombone coupled with software which responds to musical input. As this work is part of a larger project which is exploring the potential of computers to support the development of a creative approach to instrumental music, we first outline the rationale for our approach and follow with a description of the resulting music and software and its high-level design. In this paper we focus largely on the technical aspects of the project, but it should be noted that the (traditionally notated) musical composition is a critical part of the overall work and that technical development and the compositional process proceeded in parallel and strongly influenced one another.

2. THE PROJECT
As computer technology has advanced to the point where real-time audio and video processing is now viable on entry-level hardware, it is timely to consider whether intelligent use of this technology can help the development of instrumental music skills. While recording studios use digital technology routinely, its impact in *practice and teaching* studios appears generally limited. That is, the approach taken by musicians, music teachers and students to the development of musical skills has not yet been greatly impacted by technology.

This may be changing. Band-in-a-Box [1] for example has been notably successful in providing flexible backing tracks for developing improvisation skills and advanced video-conferencing technologies are being used in several music institutions to give students in remote areas access to specialist tuition. Other promising products and technologies include SmartMusic [2], which uses score-following techniques [4] to provide automatic accompaniments for soloists and Family-Ensemble [14] which uses similar techniques to allow non-expert musicians to provide full accompaniments with single-finger piano playing.

We feel that digital technologies have potential to contribute to effective instrumental music learning by providing real-time feedback to musicians on aspects of their playing, but there exist no clear, specific requirements for these types of tools. The literature on supporting creative work provides high-level guidelines (eg. [8, 11, 16]), including recommendations that the tools encourage risk-taking and avoid pseudo-objective judgments about what constitutes good playing, but specific requirements remain to be unearthed.

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 [17], 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 [10]. 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" [13]. 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 [12]. 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 [5, 9]. 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 [7] 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.

Because there are no self-evident requirements for instrumental music learning support tools, using an artistic, exploratory approach is a potentially fruitful way to uncover useful uses for technology in this area. Wilson [19] has recently examined the relationship between the arts and sciences and discussed the role of artists in exploring new technological developments. We feel that this approach is especially suited to the instrumental music domain and for this reason have chosen to use the development of an interactive music performance piece to explore the potential of computer technologies in this area. The basic idea is to bring together a composer/instrumentalist and a technologist (with a strong musical background) to collaboratively develop notated musical content and accompanying software concurrently, with the expectation that techniques which effectively link the live musician and the computer output (visual and musical) will also be applicable in designing instrumental music learning support tools.

3. APPROACH TO DESIGN AND DEVELOPMENT

Because of the exploratory nature of this work it was not possible to begin with a set of clear, unchanging requirements to be systematically implemented by following a predetermined plan. Rather, a participatory design approach was taken, meaning that software development and requirement discovery took place as part of a collaborative project between software developer (technologist) and musician (artist) [3] . 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.

Starting with simple prototype programs demonstrating some potential directions for exploration, the artist and technologist gradually developed preferences and chose aspects of the software for further refinement. Concurrently, the artist sketched musical ideas and improvised with the software to discover its capabilities and limitations.

The software development environment was “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’ [18] 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.

Both the artist and technologist worked in the same room, with a rear-projection video wall showing output from the software and a good quality audio system. The technologist could make changes to the software “under the hood” without shutting down the output in most cases, which allowed the artist to experiment with the software during this time. This capability also allowed him to see the effect of certain changes immediately, decide whether to accept or reject them and make modified requests of the technologist as appropriate.

4. THE WORK

4.1 Description

The piece is comprised of a notated piece of music for solo trombone and software written in pd. The composition (Figure 1) makes use of advanced instrumental techniques including sophisticated mute technique and multiphonics, where the musician plays one note and sings another simultaneously. Thus, the piece requires the skills of a high-level player. However, in style it does not sound “virtuosic” in that it does not contain many fast passages or explore the extreme high or low range of the instrument. Rather, the music explores a wide range of timbres and the relationship between the harmonic structure of the sound and the accompanying computer generated audio and graphics.

Figure 1. Opening section of the music for solo trombone.

The computer responds to the performance of the musician via direct audio input (two microphones), playing back animated 3D graphics coupled with computer-generated audio. We describe the visual and audio elements here separately but, as will become clear, they are in reality intimately linked.

4.1.1 Visual Elements

If there is no audio input, the computer display will appear almost completely black, but a vague outline of a string of spheres suspended from the top of the screen may just be discerned (Figure 2).

Figure 2. Diagram showing structure of computer output before any sounds have been detected. Note: the actual output at this point is very dark and the spheres barely visible.

The spheres do not remain completely stationary but move very slightly as if blowing in a breeze. The movement appears quite natural because the spheres are programmed to obey the laws of Netownian physics (see Section 4.2). The "string" linking the spheres together has a certain elasticity and so the spheres gently "bounce" around.

Each sphere is linked to a particular pitch-class or note, such as A, Bb, C, D, etc. (Figure 2), so that whenever a particular note is played, the sphere associated with that pitch-class has force exerted on it. The force is proportional to the volume of the note, so loud notes have a greater impact on the string of spheres than soft notes. A single loud F for example might cause the sphere 3rd from the bottom to rapidly move out to the left, float there for the duration of the note and then, when the note has stopped sounding, gently return to the resting position. A softer F would have the same general effect, but the sphere would move more slowly and not as far.

The harmonic structure of the sound has an effect on the appearance of the spheres. Complex sounds contain partials at frequencies other than the fundamental, and through the use of analysis objects available for pd such as fiddle~ [15], it is possible to identify the frequencies of these partials and their amplitude (or volume) in real-time. In this piece, we identify three partials in the live sound and these affect the transparency and colour of the spheres. For example, if a Bb is played, the fiddle~ pitch-analysis object may indicate that in addition to the Bb, the sound contains traces of the 3rd (F) and 5th (D) partial. Thus in our piece the Bb sphere would become the most visible (least transparent) and would glow with a bluish colour and the F and D spheres would also glow with an intensity proportional to their relative volumes (ie. less than the Bb sphere). The end result then is that one of the spheres (the Bb) is floating to one side (dragging the linked spheres with it) and glowing brightly and the D and F spheres glow also, but with less intensity. Once the note has stopped sounding, the visibility and brightness of the spheres gradually fades.

The screenshot in Figure 3 illustrates the effect. This is a screenshot of a video of a performance of the piece. It can be seen that the sphere 5th from the top (C) is glowing brightest, and, although it cannot be seen in a still picture, moving to the left, indicating that the performer is currently playing a C. Also various other spheres are glowing, either because the C being played contains partials at their frequency or because previous notes caused them to glow and they are fading. In use, it is more apparent which spheres are reacting to the current sound and which are fading, but this is difficult to convey in a still picture.

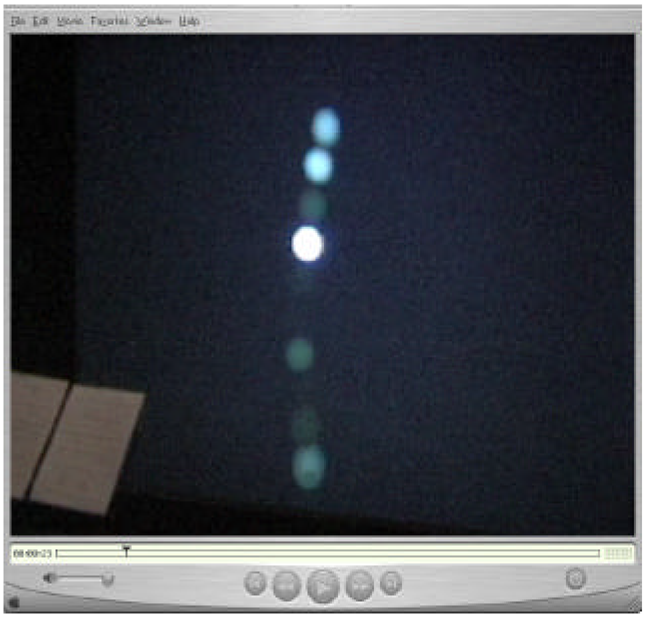

Figure 3. Video of the display during live performance. (A music stand is in the left foreground.)

It has been mentioned that a force proportional to the volume of the currently sounding note is exerted on the spheres. The direction of this force is controlled by the performer by pointing their instrument towards microphones positioned to their left or right. If they point directly to the left, the force exerted on the sphere associated with the current note pitch is (almost) exclusively to the left, if they point to the right, the force is directed to the right. The directionality actually has a kind of 3D effect, because if the input level is equal from left and right microphones then the force is directed in the '+Z' direction (ie. inwards, away from the performer). If the input is slightly to the right, then the force is exerted proportionally in both the +Z direction and towards the right (the '+X' direction). The effect is that the performer can "push" the spheres around by pointing their instrument while playing.

4.1.2 Audio Elements

In addition to visual feedback, the software produces audio influenced, but not controlled, by the musical input. We have previously explained how the sphere that force is exerted upon is selected by the pitch-class of the note being played. By definition a pitch-class refers to all notes with the same name and not to a note with a specific frequency. Thus the pitch-class 'A' refers to all A's in all octaves, not to a specific A such as the one above middle C (440 Hertz). When the software detects an A, whether a partial or not, it makes a note of its frequency and associates that frequency with the A sphere. It will then play a cosine wave (pure tone) with that frequency at a volume proportional to the current velocity of that particular sphere. This means that while any sphere on the screen is moving at a speed greater than an adjustable threshold value, the system is generating sounds.

For example, suppose the musician plays an A above middle C, which has a frequency of 440 Hertz. The A sphere will 'store' the value of 440 Hertz and play a cosine wave with that frequency whenever the A sphere is moving. The faster the sphere moves the louder the 440 Hertz cosine wave note will sound. In addition, the frequencies up to three partials detected in the musician's A will be stored also. If, for example, the played A contained the E partial with a frequency around 1319 Hertz, then the E sphere would store this frequency and play it back. Once again, the faster the E sphere moves the louder that frequency will sound.

It should be noted that as the software is configured, each sphere can only store one frequency, so that if the musician were to play another A with a different frequency then this would replace the existing frequency immediately, and sphere movement would then cause the new frequency to be output.

The effect is difficult to describe and can vary depending on various sensitivity settings of microphones, etc., but it could be described as a kind of synthesis of the musician's sound mediated by the physical structure of the string of spheres. Because the software detects some aspects of the harmonic structure of the musician's sound, the playback from the computer can sound a little like an echo, strangely mutated but somehow not unnatural. The system is complex enough that no two performances are alike, yet the musician has control over the overall direction and sound without having precise control over the details. In a sense, the system acts as an extension of the instrument, transforming the sound in ways that might be sometimes surprising but still retaining controllability.

4.2 Software design

We have attempted to modularize the software into a kind of tiered design in the expectation that future projects involving the creation of similar computer-based "sound sculptures" can use the same basic architecture. Pd uses a visual programming approach in which user-interface elements are incorporated with representations of the objects used to implement functionality. This has advantages and disadvantages, but one bonus is that the high level structure of a program can be made apparent fairly easily. Figure 4 shows the main screen of our application. Without going in to detail of how pd works, it can be seen that at a high-level our application is comprised of five modules (the rectangular objects in the screenshot): spheres-audio-analysis-main (audio analysis), spheres-model-force-manager (force manager), spheres-ui-output-manager (output manager), spheres-ui-window-manager (window manager) and spheres-audio-processor-main (audio processor).

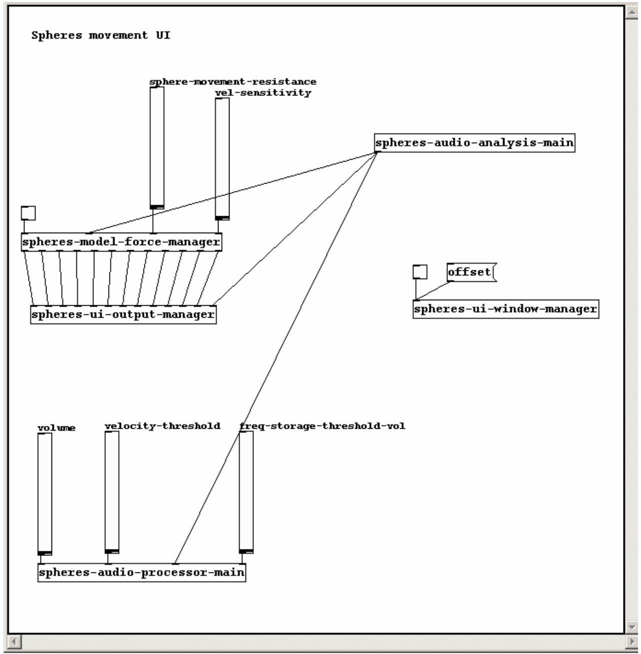

Figure 4. View of the main screen of the application, showing high-level structure.

The flow of data between the modules is indicated by the lines linking them together. Thus it can be seen that the audio analysis module, which analyses the input from the microphones and determines the pitch, harmonic structure and stereo position, sends this information to the force manager, the output manager and the audio processor.

The force manager and output manager are closely linked, with the force manager converting the volumes of harmonics into forces which are applied to the physical model of the spheres. This module makes extensive use of the "Physical modeling for pure-data" (pmpd) module developed by [6]. Using pmpd, the developer can easily set up a complete model of a physical system made up of objects joined together by links of given elasticity. An example patch supplied with pmpd provided the initial inspiration for the visualization used in our software.

The output manager takes the output from the physical model, which is the position of all the spheres taken every 10ms, and displays them on the screen in 3D space using another module for pd, the Graphics Environment for Multimedia (GEM) which provides powerful 3D graphics capabilities. It should be noted that this module also receives data from the audio analysis module so that it can control the colour and transparency of the spheres based on the harmonic structure of the sound.

The audio processor receives information from the audio analysis module so that it can associate (or 'store') the pitch frequencies contained in the musician's sound with the appropriate spheres. In addition (although it is not visible in this figure) it receives the velocities of each sphere from the force manager, as these control the output volume of each stored frequency.

Finally, the spheres-ui-window-manager module is simply responsible for opening and closing the output window.

The sliders in the figure (volume, sphere-movement-resistance, etc) are needed to adjust the sensitivity of various aspects of the system. Depending on the placement of the microphones, various characteristics of the particular installation site and the preferences of the individual musician, it is necessary to adjust:

- how responsive the spheres are to audio input;
- the ratio between sphere movement and output volume;
- the lowest sphere velocity which causes audio output;
- how loud a detected pitch must be before it is stored for later playback; and
- the overall output volume.

5. DISCUSSION

The rationale for this project was to develop a multimedia performance piece which explores the potential of real-time digital audio/visual processing software to provide meaningful feedback to musicians on their playing. We feel that the resulting software has potential to be useful to musicians and music students in several ways. Firstly, it is a piece of music for solo trombone which makes interesting use of new technology. This piece could be seen as a kind of "multimedia etude" for trombonists in that while it explores certain technical aspects of the trombone, namely the harmonic structure of its sound and legato (smooth) playing, it stands alone as a piece of music and is not merely an exercise.

In addition, the software environment can be used on its own for musicians to experiment with. We believe that the richness and complexity of the software can provide a stimulating environment for musical experimentation and improvisation.

It is also possible that this software could be used in teaching situations. For example, teachers could demonstrate some aspect of instrumental technique and ask the student to note the effect that their playing has on the computer display and/or audio output. The system could provide an alternative way for students to perceive the difference between the teacher's playing and their playing and thus encourage them to notice aspects of their sound or musicianship that they might not detect aurally.

The project has also provided a framework for the development of both further artworks and software which is more specifically targeted at particular technical issues faced by musicians. In particular, we feel that the physical modeling techniques used in this project have great potential for providing feedback to musicians on various characteristics of their playing in ways that are intuitively understandable.

6. CONCLUSION AND FUTURE DIRECTIONS

We have described a project which resulted in the development of a multimedia artwork exploring the capabilities and possibilities of digital technologies for providing feedback to musicians. We have outlined why an 'artistic' approach was taken and detailed the features and design of the resulting software.

This work has provided a solid foundation for further development. Future work is likely to involve evaluation of the existing software by a number of musicians to help identify characteristics they find useful and those which are less helpful. Building on this, we expect that a series of tools, focused on particular aspects of instrumental technique, but still encouraging a non-judgmental, creative approach will be developed utilizing the technologies described in this paper.

7. ACKNOWLEDGMENTS

Our thanks to the developers of Pure Data, Physical Modeling for Pure Data and the Graphics Environment for Multimedia for providing the tools to make this work possible.

8. REFERENCES

[1] PG Music Inc. - Band-in-a-Box 2005 for Windows, http://www.pgmusic.com/bandbox.htm

[2] SmartMusic, http://www.smartmusic.com/

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

[4] Dannenberg, R. Real-time scheduling and computer accompaniment. in Matthews, M. and Pierce, J. eds. *Current directions in computer music research*, The MIT Press, Cambrige, MA, 1989, 225-261.

[5] Dixon, S., Goebl, W. and Widmer, G., Real Time Tracking and Visualisation of Musical Expression. in *Music and Artificial Intelligence: Second International Conference*, (Edinburgh, Scotland, UK, 2002), Springer-Verlag Heidelberg, 58-68.

[6] Henry, C., Using Physical Modelling for Pure Data (pmpd) with an audio and video synthesis, http://puredata.info/community/projects/convention04/lectures/tk-chenry/pmpd.pdf

[7] Johnston, A., Amitani, S. and Edmonds, E., Amplifying Reflective Thinking in Musical Performance. in *Creativity and Cognition 2005*, (London, 2005).

[8] Johnston, A. and Edmonds, E., Creativity, Music and Computers: Guidelines for Computer-Based Instrumental Music Support Tools. in *Australasian Conference of Information Systems*, (Hobart, Tasmania, 2004), School of Information Systems, University of Tasmania.

[9] Langner, J. and Goebl, W. Visualising Expressive Performance in Tempo-Loudness Space. *Computer Music Journal*, *27* (4). 69-83.

[10] Miller, D.G. and Schutte, H.K., Documentation of the Elite Singing Voice, http://www.vocevista.com/contents.html

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

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

[13] Oliver, W., Yu, J. and Metois, E., The Singing Tree: Design of An Interactive Musical Interface. in *Proc. Designing Interactive Systems*, (Amsterdam, The Netherlands, 1997), ACM Press New York, NY, USA, 261-264.

[14] Ooshima, C., Nishimoto, K. and Suzuki, M., Family ensemble: a collaborative musical edutainment system for children and parents. in *12th annual ACM international conference on Multimedia*, (New York, NY, USA, 2004), ACM Press, 556-563.

[15] Puckette, M.S., Apel, T. and Zicarelli, D.D., Real-time audio analysis tools for Pd and MSP. in *International Computer Music Conference*, (San Francisco, 1998), International Computer Music Association, 109-112.

[16] Shneiderman, B. Creating Creativity: User Interfaces for Supporting Innovation. *ACM Transactions on Computer-Human Interaction*, *7* (1). 114-138.

[17] Thorpe, W., Visual feedback of acoustic voice features in singing training, http://www.med.rug.nl/pas/Conf_contrib/Thorpe/Thorpe_bio_touch.htm

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

[19] Wilson, S. *Information Arts: Intersections of Art, Science and Technology*. MIT Press, Cambridge, Massachusetts, 2002.

The Second

Australasian Conference on Interactive Entertainment

University of Technology, Sydney, Australia

23-25 November 2005

Edited by

Yusuf Pisan

Sponsored by

Australian CRC For Interaction Design

Australian Film Television Radio School

La Trobe University

in cooperation with

SIGART, SIGCHI, SIGGRAPH and SIGWEB

Creativity and Cognition Studios

Creativity & Cognition Studios Press
University of Technology, Sydney
PO Box 123, Broadway
NSW 2007
Australia
Telephone: +61 2 9514 4640
Fax: +61 2 9514 4492
URL: http://www.creativityandcognition.com/

ISBN 0-9751533-2-3

Printed by UTS Printing Services
Cover Design by Viveka Weiley