

etherSound - an interactive sound installation

Henrik Frisk

Malmö Academy of Music - Lund University

Box 8203, 20041 Malmö, Sweden

henrik.frisk@mhm.lu.se

7th February 2005

Abstract

This article describes the interactive instrument/sound installation etherSound and discusses its artistic and ethical implications. etherSound is a work in progress and the main intention is to create a vehicle for audience participation through the use of SMS (Short Message Service). The two different contexts in which etherSound has been tried (in concert with performers and as a sound installation without performers) is discussed as well as the design of the system and the mapping between text and sound. A notion of a 'democracy of participation' is introduced. The relatively fast response of the system, the familiarity of the interface (the cellular phone) and the accessibility of the system suggests that the cellular phone can be successfully integrated in a sonic art work.

1 Introduction

etherSound was commissioned by the curator Miya Yoshida for her project *The Invisible Landscapes* and was realized for the first time in August 2003 at Malmö Art Museum in the city of Malmö, Sweden. The curatorial concept for *The Invisible Landscapes* project was the use of cellular phones in the context of experiencing and creating artistic expressions. The principle idea behind etherSound came to be an attempt at developing an instrument that can be played by anybody who has knowledge about how to send an sms (Short Messages Service) from their cellular phones. The focus of my research project, of which etherSound is a part, is interaction between computers and musicians as well as non-musicians. etherSound is an investigation of some of the aspects of interaction between the listener, the sounds created and the musicians playing, and also of the formal and temporal distribution of the music that this interaction results in.

While interaction is an important aspect of musical performance in many genres, active audience participation is not as evolved in the western music tradition, and when explored, the result is usually not labeled as music, but rather as a sound installation, soundscape, sonic art or some other term that indicates alienation from the traditional notion of music. Opening up a musical work for others than trained musicians is not a trivial task; careful attention has to be paid to the purpose of doing so and to the intentions of the work. It is relevant to pose the question whether it is possible to reach a satisfactory result with almost no limitations on participation and, if so, can the result not be called music? However, before these questions can be addressed we need delineate the purposes for wanting to allow for public participation.

Public participation has been explored in the visual arts for almost a century, for artistic as well as political reasons, and if we look at it from a performing arts perspective, the audience visiting a performance can be said to participate in it - if only in a limited sense. Concurrently, especially in spheres of distribution and consumption of music, there is a tendency to objectify the musical work. As the power and irrational control exercised by the institutions of distribution increases, the freedom of choice and influence of the listener decreases [Adorno,]. Furthermore, western art music is to a considerable extent looked upon as a hierarchic process; a process that begins in the mind of the composer and ends at the level of the listener or, even before that, at the level of interpretation. It is fair to assume that bringing in an uncontrollable agglomeration of participants influencing the distribution of musical events will disturb this order .

In their article on the multi-participant environment *The Interactive Dance Club*, Ulyate and Bianciardi defines one of the design goals as wanting to ‘deliver the euphoria of the artistic experience to “unskilled” participants’ [Ulyate and Bianciardi, 2002]. Instead of sharing merely the result with an audience, they attempt at unfolding the creative process leading to the result and invite the audience to take part in this process. This ambition points to another issue: how to design musical interfaces that have a ‘low entry fee, with no ceiling on virtuosity’ [Wessel and Wright, 2002, Jordà, 2002] (see also [Rowe, 1993, Freeman et al., 2004]). With the recent technological advances there are innumerable tools that can be used for collaborative efforts [Barbosa and Kaltenbrunner, 2002], affordable devices that easily can be used as interfaces to computer mediated art works. Not only has this the potential of changing our perception of the arts, it can also help us understand and this new technology and the impact it has on our lives.

Traditionally there is an intimate association between social class, level of education and cultural interests [DiMaggio and Useem, 1978, Bourdieu, 1979] that affects cultural consumption. Is it possible to make music that can counteract this ‘closeness’ of contemporary art and music; that can make conditions for classless and unprejudiced participation

in the arts without compromising the content and the expression? I believe it is and I believe collaborative music is one way to achieve this. Roy Ascott, in addressing the issue of ‘content’ in art involving computers and telecommunications writes:

In telematic art, meaning is not something created by the artist, distributed through the network, and *received* by the observer. Meaning is the product of interaction between the observer and the system, the content of which is in a state of flux, of endless change and transformation [Ascott, 1990].

Following this line of thought, it may be concluded that the need for a thorough insight in the history of art or electronic music is no longer a prerequisite for understanding a collaborative, interactive work. This limits the advantage of the educated listener and makes room for new interpretations of the term ‘understanding’ in the arts.

2 The Design

etherSound is an attempt to open a musical work to the uninitiated and provide for a notion of ‘democracy of participation’: all contributions are equally valuable. Accessibility without prior knowledge of music or musical training is an end in itself in this project. It should be noted that this obviously presupposes that the participant knows how to send a sms and that the system makes it difficult for those who are not familiar with this technology¹. It should also be made clear that, using sms text messages for interaction as it is implemented here, does not allow for direct dynamic control. Every message generates one ‘message-composition’ and all control data is derived from the content of the message.

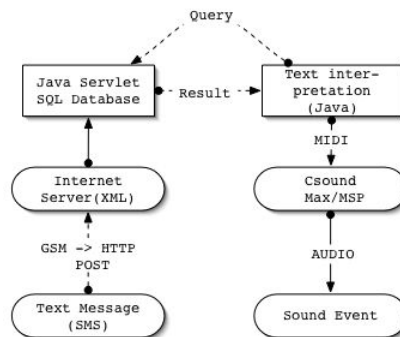


Figure 1: Communication in the first version.

2.1 The first model

In the first version, realized in August 2003, the communication between the participant and the system was accomplished according to Figure 1. A sms sent to the specified number was transformed to a XML file and transferred to a URL by a HTTP POST request. This part was handled through an external service. At the called URL, a JSP (Java Server Pages) was directing the POST data to a Java Bean [Java Enterprise Edition, 2004] that

¹Yet there is a great commercial interest in increasing the use of sms and, in Sweden, there has been a tremendous effort from the part of the GSM service providers to teach their customers how to use it.

handled the parsing of the data and the connection to a MySQL database in which it created a new entry with the relevant fields.

It was due to security reason at the museum where this version was realized that the HTTP request could not be handled locally. Instead, the local computer queried the server database for new entries on regular intervals. After some testing, sending a SQL query once every second seemed like a reasonable time interval. Shorter time intervals didn't accomplish a perceivably quicker response time and, since the synthesis program was running on the same machine, I didn't want to use more processing and network activity than necessary for this task (see section 3 for further discussion). After the text message had been processed, control signals were sent by MIDI to the synthesis engine.

2.2 The current model

Although the first version worked well and was fairly stable, it was a solution that required an external sms processing service, and a local, reliable network connection. In order to make the piece more 'portable' and independent, the message receiving part has been rebuilt. Using the gnokii API [gnokii, 1995] it becomes relatively easy and reliable to connect a GSM phone to a computer and thus enable reception of the sms messages locally. To have the possibility to review the activity of transmission, the messages are, just as in the first model, written to a database. In other words, the client-server model is retained but on one and the same machine. Furthermore, the MIDI connection between the control application and the synthesis engine has been replaced with OpenSound Control (OSC) [Wright et al., 2003, OSC, 1997] for speed, reliability and flexibility, using the library JavaOSC (see <http://www.mat.ucsb.edu/~c.ramakr/illposed/javaosc.html>).

2.3 The text analysis

The program handling the text processing and the mapping of text to control signals for the sound synthesis is written in Java [Java Standard Edition, 2004] and features a simple but useful GUI for control and feedback about the status of the system. It is here, in the mapping between the text and the sound, that the compositional choices have been made. There are three groups of parameters that are being extracted for every message:

- The length of the whole event
- The rhythm and articulation of the individual sound events
- The pitch and character of individual sound events

For the timing there are two parameters; a local 'life' index shaping the rhythms and the length of the current message and a global index that influences the current and subsequent 'message-compositions'. The global index is a function of the current and previous messages local indexes. The purpose of the local index is to make a simple semantic analysis of the message and discriminate between a set of random letters and real words. The participant should be 'rewarded' for the effort of writing a message with substance. The local index is calculated by looking at the average length of words and the average number of syllables per word and comparing these with constants:

$$i_1 = \frac{1}{(w(\frac{c}{w_c}) - w_l)^{1/2} + 1} \quad i_2 = \frac{1}{(w(\frac{s}{w_c}) - s_l)^{1/2} + 1} \quad (2.1)$$

Table 1: Life index for four different messages

<i>message</i>	<i>life index</i>
hello	0.18882
<i>Hello, my name is Henrik</i>	<i>0.81032</i>
hjdkls la s duyfke jhslfd hasdfiw uehr jkdsf	0.14448
<i>From fairest creatures we desire increase, That thereby beautys rose might never</i>	<i>1.44618</i>

where c and s are the total number of characters and syllables, w_c is the number of words in the current message, w_l and s_l are constants defining the ‘optimal’ mean number of words/syllables. w is a weight defined by

$$w = \frac{1}{w_c - s_c + 0.5} \quad (2.2)$$

where s_c is the total number of words that contains vowels. Through w , the index is decreased if the message contains words without vowels. The mean value of i_1 and i_2 is then multiplied by the arcus tangens of the number of words in relation to a third constant parameter, o_w , delimiting the optimal number of words per message² according to (2.3).

$$lifeIndex = \frac{i_1 + i_2}{2} \arctan\left(\frac{w_c}{o_w}\right) \quad (2.3)$$

If we set w_l to 4.5, s_l to 2.0 and o_w to 10 the result on four different messages can be seen from Table 1; the method distinguishes fairly well between nonsense and real words at a low cost. Similar or better results could conceivably be achieved in a number of different ways but this method appears to work well for the purpose. Since there is only audio feedback, it is important that all, even empty messages, will lead to a perceptible change in the sonic output. The total length of the music derived from the message is a function of the local index. Any new messages received adds its local index to the instantaneous global index which constantly decreases exponentially at a set rate³. If a message causes the global index to reach maximum, it stops the playback of the current message and begins playing back a pre-composed pattern, sonically different from the output of a typical message, for about 30 seconds before resuming ordinary mode and starts playing back the message that caused the break. This feature is added to reward collaborative efforts. The global index controls mainly the density and the overall volume of the output, but also the distribution of random and stochastic processes in the synthesis.

Every word of the message generates one musical phrase. The duration of each phrase is determined from the number of syllables in the originating word. Punctuation marks bring about rests.

2.4 The synthesis

The synthesis engine is written as a Csound orchestra [Boulanger, 2000] (see also <http://www.csounds.com/>) running inside a Max/MSP (<http://www.cycling74.com/products/maxmsp.html>) patch through the use of the `csound~` object (see <http://www.csounds.com/matt/>). The ‘score’ for the message to be played back is sent to Max/MSP using OSC. Max/MSP is responsible for

²Since a SMS is limited to 160 characters these constants are set according to what kind of message content should be rewarded.

³The rate is context dependent. In a performance with improvisation it would be shorter than in an installation.

timing the note events and preparing valid information for the `csound~` object and the orchestra file associated with it. Due to processing power limitations only one message can be played back simultaneously; if a message is received before the previously received message is finished playing back, the new message will interrupt the first message.

All sounds heard in etherSound are generated with FOF (Fonction d'Onde Formantique) synthesis as this technique is implemented in Csound [Clarke, 2000, Byrne Villez, 2000], using both samples and simple sine waves as sound sources. There are two distinct timbres in each 'message-composition': one is a bell like sound whose timbre is governed by the series of vowels in the text. This sound has three or more voices and creates the harmonic progression. The pitches are mapped according to the occurrence of certain key letters in the originating text⁴. After the initial chord has been introduced, all voices begin a glissando toward the centre between the outer limits of the chord, creating microtonal variations of an ever decreasing harmony, ending at a unison. This voice is a horizontal contrast to the second voice.

The second voice uses samples of a male reading a text in english⁵ as its sound source to the FOF opcode. From this recording, short sample buffers (typically 4096 samples) has been extracted, one for each letter. The letters in the message are mapped one to one to these samples. In this voice the FOF synthesis is used to granulate the samples, thus creating an erratic, non-tonal texture under the harmonic bell-like sounds described above.

3 Discussion

The latency of the system in the first model, measured from when the participant presses the 'send' button to when sound is heard, is in the range of less than a second to a little over two seconds. This may seem long but, in fact, many users commented the fact that they experienced the response time as being short. The second model remains to be tested, but it is fair to assume that the response will be slower.

etherSound has been used in two different contexts. As a stand alone sound installation that users can interact with but also in combination with one or several improvising musicians playing acoustical instruments. In this situation, which resembles a traditional concert, the audience is 'playing' the electronic instrument and are given an important role in the development of the form. As can be gathered from the description of the system above, the sonic outcome of a received sms is fairly strictly controlled. On the individual level, only a limited amount of control over detail is offered, and it is debatable whether etherSound can be called an 'instrument' at all. This was however never the intention. It is the *desire* to contribute to the whole that was intended to be the ruling factor, not the individuality of expression or the virtuosity of performance. Thus, etherSound is closer to a musical composition with a stochastic process ruling the distribution of sonic events.

An interesting aspect of the concert performance context appears if we compare it to an interactive performance for computer and instrument where the performer influences the output of the computer. In this model the performer and the computer constitute an ontological entity, a closed system that the audience can observe and listen to. However, in etherSound, the computer generated sounds becomes the common ground between the performers and the audience, a sonic field of communication and the audience can no longer be disunited from the content.

Whether or not the participants felt they had influence and whether this influence set

⁴For the sake of experiment and variation, I am changing these 'key notes' for every performance of etherSound.

⁵An excerpt from the recording of one of John Cage's lectures at Harvard College 1989.

creative energies in motion within the participant can only be proved, if at all, by performing empirical studies that are beyond my intentions and competence. I can however offer the lightweight, subjective analysis that improvising along with an audience in a way that can be done with this work, is an experience incomparable to traditional group improvisation.

4 Future improvements and further work

The aspect of ‘democracy of participation’ could be further expanded by also streaming the sound on the Internet, inviting participants anywhere to collaborate. It would also be desirable to allow for simultaneous playback of multiple messages, possibly through the use of several computers, and to add more depth to the interface and allow for ‘expert’ performers. One thought is to add the possibility to make a voice call to the phone connected to the system and interact in true real time, either by voice or by pressing digits. The text analysis responsible for calculating the life index could be further evolved, i.e. to allow for, and equally reward, typical sms language such as ‘c u 4 dinner 2nite’.

Since every performance of etherSound is ‘recorded’ in the database, the music can be recreated and altered. I am currently working on a fixed media piece using the data collected during one of the performances of etherSound.

5 Acknowledgments

I wish to thank Miya Yoshida who commissioned etherSound, Leif Lönnblad for his work, Lund University/Malmö Music Academy for financial support and Vodafone™ for financial and technical support in the beginning of the project. Furthermore, I wish to acknowledge the people behind the *gnokii* project and the people behind the library *JavaOSC*.

References

- [Adorno,] Adorno, T. W. chapter 12, page 211. In [Adorno, 1962].
- [Adorno, 1962] Adorno, T. W. (1962). *Inledning till Musiksociologin (Original title: Musiksoziologie)*. Bo Cavefors Bokförlag.
- [Ascott, 1990] Ascott, R. (1990). Is there love in the telematic embrace? *Art Journal - Computers and Art: Issues of Content*, 49(3):241–247.
- [Barbosa and Kaltenbrunner, 2002] Barbosa, A. and Kaltenbrunner, M. (2002). Public sound objects: a shared musical space on the web. In *Proceedings. Second International Conference on WEB delivering of Music (WEDELMUSIC'02)*, pages 9–16. IEEE, IEEE.
- [Boulanger, 2000] Boulanger, R., editor (2000). *The Csound Book, Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*. MIT Press, 2 edition.
- [Bourdieu, 1979] Bourdieu, P. (1979). *Distinction: a social critique of the judgement of taste*. Harvard University Press. Translation by Richard Nice.
- [Byrne Villez, 2000] Byrne Villez, P. (2000). *Processing Samples with Csound's FOF Opcode*, chapter 15, pages 307–320. In [Boulanger, 2000], 2 edition.

- [Clarke, 2000] Clarke, M. (2000). *FOF and FOG synthesis in Csound*, chapter 14, pages 293–306. In [Boulanger, 2000], 2 edition.
- [DiMaggio and Useem, 1978] DiMaggio, P. and Useem, M. (1978). Social class and arts consumption: The origins and consequences of class differences to the arts in america. *They and Society*, 5(2):141–161.
- [Freeman et al., 2004] Freeman, J., Ramakrishnan, C., Varnik, K., Neuhaus, M., Burk, P., and Birchfield, D. (2004). Adaptive high-level classification of vocal gestures within a networked sound instrument. In *Proceedings of the International Computer Music Conference 2004*. ICMA.
- [gnokii, 1995] gnokii (1995). Web page. <http://www.gnokii.org/index.shtml>.
- [Java Enterprise Edition, 2004] Java Enterprise Edition (2004). *J2EE, API Specification*. Sun, <http://java.sun.com>, 1.4 edition.
- [Java Standard Edition, 2004] Java Standard Edition (2004). *J2SE, API Specification*. Sun, <http://java.sun.com>, 1.4.1 edition.
- [Jordà, 2002] Jordà, S. (2002). Fmol: Towards user-friendly, sophisticated new musical instruments. *Computer Music Journal*, 26(3):23–39.
- [OSC, 1997] OSC (1997). Web page. <http://www.cnmat.berkeley.edu/OpenSoundControl/>.
- [Rowe, 1993] Rowe, R. (1993). *Interactiv Music Systems: Machine Listening and Composing*. MIT Press.
- [Ulyate and Bianciardi, 2002] Ulyate, R. and Bianciardi, D. (2002). The interactive dance club: avoiding chaos in a multi-participant environment. *Computer Music Journal*, 26(3):40–49.
- [Wessel and Wright, 2002] Wessel, D. L. and Wright, M. (2002). Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22.
- [Wright et al., 2003] Wright, M., Freed, A., and Momeni, A. (2003). Opensound control: State of the art 2003. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, pages 153–159, Montreal, Canada. NIME-03.