The Intelligent Street: responsive sound environments for social interaction

Henrik Lörstad
Sonic Studio
Interactive Institute
Sonic Acusticum 4
941 28 Piteå
Sweden
henrik.lorstad@tii.se

Mark d'Inverno
University of Westminster
School of Computer Science
115 New Cavendish Street
London W1M 8UW
United Kingdom
dinverm@wmin.ac.uk

John Eacott
University of Westminster
Department of Music
Harrow Campus
London HA1
United Kingdom
eacottj@wmin.ac.uk

ABSTRACT

The *Intelligent Street* is a music installation that is able to respond intelligently to the collective requests of users interacting together. The performance it creates is largely influenced by the collective set of text commands from users' mobile phones. In this way, users in shared environments, subjugated for so long to uncontrollable and often undesired 'Muzak', can now directly influence their sonic environment and collectively create the aural soundscape that *they* desire. We see our project as enabling inhabitants of any given space from *passive consumers* to *active creators*, and anticipate it has significant commercial, social and educational potential.

In this paper we present a description of the installation, its software architecture and implementation, as well as a report on subsequent user-evaluation in providing a musical public playground and, moreover, our over-arching goals as musicians and software engineers.

Categories and Subject Descriptors

Sound and Music [**Social Computing**]: New Genres

General Terms

Ambient Intelligence, Intelligent Responsive Sound, Algorithmic and Generative Composition, Software Architectures, Human Computer Interaction

Keywords

Algorithmic Music, Intelligent Responsive Sound, Prototype

1. INTRODUCTION

The use of music (or Muzak!) in public places is extremely common. However, any individual within that space has absolutely no control over the performance of that music, and it can often become intrusive and unwelcome. In our work we are interested in building interactive and responsive sound installations, where the performance results directly from the interaction of the users within a given space. This project, known as the Intelligent Street, has been developed collaboratively in Sweden and the UK.

1.1 Background

This project was based in part upon the success of *the street* project, an interactive sound installation, designed by Eacott, that ran at the University of Westminster in September 2000. Its aim was to create music in response to the activity of staff and students passing through the entrance of the University of Westminster. Ultrasound sensors at each end of the street were used to register the amount of human activity. As the activity increased so did the energy of the music played.

However, the basic algorithms for responding to the user were quite simple and the user had no direct control over the output. In collaboration with d'Inverno, experienced in the field of intelligent agents [3, 8], the Intelligent Street was conceived as an installation where users could interact with each other and the installation more explicitly [5]. The medium of communication that we chose for interacting with the installation was mobile phones and text messaging in particular. By this, we are simply acknowledging the popularity of this type of communication in today's society.

The project was located in the same location (known locally as the street) in Westminster, but extended by having a parallel installation running at the Interactive Institute in Piteå, Sweden. The corresponding space is situated in the lobby of the Acusticum building. The two installations are connected via the Internet and a web portal allows users to monitor the activity in either location by the sequence of text messages.

The ideas of Eacott and d'Inverno were developed by the Ambigence Group, which includes musicians, computer programmers and artists based in the UK and Sweden developing projects which fuse arts, mostly music, with techniques from artificial intelligence.

1.2 Project Overview

The Intelligent street is a practical attempt to explore some of the ideas outlined above. Pre-composed pieces of music, written algorithmically, by Olofsson, Lörstad and Eacott are restructured, combined, manipulated and then processed as determined by users' commands. Essentially, text messages are translated into musical operations. Subsequently, users text commands affect the mood, energy and style of the music and thus the installation reflects the desires of those around it.

The overall output of the installation is thus the result of the history of dialogue created by the commands of the users. In this way we were able to to create a flexible sonic environment in which users would interact to build their shared musical soundscape. One of our goals was to demonstrate the social need for aural as well as visual stimulation within a well designed environment that could impact on future architectural designs.

At present we use various statistical data to help us understand user patterns in order to tweak IStreet's performance. The frequency of different commands is logged, for example, and this information can be used to register the best working combinations. In time, this data log will provide useful information about the tastes and wishes of users, and inform future design decisions about similar sound installations in general.

The mobile phone has been selected as the controlling medium for this project because of its widespread use as a tool for communicating today, available to practically everyone. By making use of this easily accessible tool we offer almost passerby the opportunity to actively engage in affecting the sonic environment. Intelligent Street seeks to explore new possibilities and unexpected applications for the mobile phone. It is a project in which we wish to explore ways of composing music for non-linear media and attracting participation in a creative process through interaction.

1.3 Related Work

As we understand it, this is the first work of it's kind to use algorithmic music as the medium for enabling explicit user control over an environmental sound installation. Of course there are countless mechanisms for providing environmental or 'ambient' sound, but they are typically more closely entwined with the 'natural' environment rather than the artificial one created in the world of text messaging. Perhaps the most common example is the wind-chime, used across many cultures, and still a magical way of using the natural environment to create a musical performance. A less well known *suikinkutsu*, a device used in Japanese gardens which allows water to drip onto a resonating chamber. It is argued that the occasional sound of dripping emphasises the relative silence of the natural environment in-between [13]. A sound installation which used kinetic mechanisms based on wind power include Max Eastley's work the *Bamboo Circles* [6].

Contemporary algorithmic-based examples of providing sonic environmental pieces include Jem Finer's 1000 year composition *Longplayer* installed in a lighthouse in Wapping, London [7]. A piece of music that the composers guarantees will not repeat itself once during its (quite substantial!) life. There is also the furniture designed in the *Unfoldings* project that is based on ideas similar in spirit to our own [1] of creating more responsive, non-naturally occurring environments. In this physical space is a collection of furniture: pillows, armchairs and rugs. Depending whether users stand, sit or lie, *Unfoldings* responds in sound, music and light.

As we mentioned above we are not aware of examples of sound art using SMS interaction although we are aware of a few related projects. specifically using handheld technology, was demonstrated by the SSEYO/Tao group, who performed live generative music using a palm top running *Intent Sound System* [12].

1.4 SuperCollider

SuperCollider2 [11] (SC) was chosen as the synthesis platform for the main audio engine of the installation. Over its 8 years of development SC has established itself as probably the most powerful, flexible and stable real-time synthesis environment available. The downside of using it is that as a code-based environment it is less easy to use by those without Object Oriented Programming experience than say Max/MSP [14]. It was decided that it would be safer to continue development in SuperCollider2 (SC2) rather than SuperCollider3 (SC3) that was also available and being developed as open source as the implementation documentation of SC3 was not complete. Along with the SC2 sound engine there were graphics, video-link and the connection to the Vodafone SMS services which were handled by custom written software in Max/Nato.0+55 and Python. The communication between the programs is written using the Open Sound Control (OSC) protocol.

1.5 The IStreet Installation

The physical environment is a walk-through area with soft, barely audible background sound. The pre-defined commands available to users are highlighted in scrolling text on a projected graphic display within the walk-through area. In this project we did not give details of precisely what the commands did and part of our evaluation strategy has been to ascertain whether the affects of a command satisfied the users expectation or intention. The user sends a text message and it is stored in a buffer. At regular intervals all the commands in the buffer are used to determine how the music alters, and the buffer is cleared. Thus, the music changes instantly (i.e. discontinuously), and a new aural experience is created. As with all human action, one of our primary considerations was that users should be able to clearly notice the effect their interaction has produced on their environment. Additional commands can tailor the music or alter it radically. If user activity reduces the music gradually decreases in volume and if no-one interacts for a long enough period the whole installation goes back to the barely audible background mode.

Along with the menu of available commands, the visual projection shows a live video feed of the equivalent space in London or Piteå respectively. Text projected on this image informs the user about the interactions taking place. All user interaction is logged and when a command is sent it is projected on the image together with an abbreviation (so that users do not reveal their whole number) of the sender's

phone number. The user will be able to see from who each command originates and at what time it was sent and relate this to the changes in the music.

The are also visual aspects of the piece that make up an integral part. Through these we can see which command is currently influencing the music. Pending commands are also displayed, all of this designed to improve comprehension and increase peoples willingness to take part in the installation. The participants are then given a sense of excitement and expectation about how their commands may effect the performance. It also indicates to other uses how they want to change the music and this can result in the users talking to each about what they want. The commands are each associated with a timer that counts down to 0, when the command is processes and the performance changes accordingly. What is interesting then is how satisfied the user is about what actually happens. Clearly, for this system to work in general there must be some sense of satisfaction or at least of pleasant surprise at emergent qualities of the music. (There is also a practical problem here in that there is always a time delaying with SMS messaging and therefore it would be very difficult to have instantaneous responses to the music.) If no more commands are input then the music will decay over a period of around a minute - and the video-link gets dimmed.

These commands determine how the musical content and structures of IStreet are altered. Every 16 bars queued text commands are processed and the new musical performance is output. If no commands in the buffer are available for processing then the music will continue playing its next logical phase. For example, the performance may move from a 'verse' structure to a 'chorus' structure. In this way the music still progresses with drum fills, chord progressions and melodies even without user intervention. The musical output is derived from a set of pre-existing musical styles and the ongoing history of all user text commands.

The actual commands available may change periodically as the music in the system is extended with help from composers from the Academy of Music in Piteå and the University of Westminster. Indeed, in future incantations of this project we would hope that the language for interacting with the installation would develop organically through use rather than being static and imposed by us as system designers.

In addition to the visual connection an aural connection between the remote cites was also set up. In Piteå the audience was given the option of using headphones to listen to musical journey happening in London. The music from the Westminster installation was streamed and sent via a separate TCP/IP-connection to a computer in Piteå. Part of the reason was to encourage music students from each location to listen to what was being created by their counterparts from a different culture. We hoped that this would stimulate musical ideas being exchanged between students in Sweden and the UK.

The visual aspects of the piece make up an important part. Through these we can see which command is currently influencing the music. Pending commands are also displayed, all of this designed to improve comprehension and increase peoples willingness to take part in the installation and we gratefully acknowledge their support.

Since one of the ideas with this project was to make it as commonly available as possible we did not want users to be hindered in experimenting with the piece by the costs of sending text messages. The Intelligent Street is best experienced and understood when given some time interacting with it. Vodafone supplied a number of free phone cards for us to offer the users wanting to interact without any costs.

2. SOFTWARE ARCHITECTURE FOR MULTI-USER INTERACTION

2.1 System description

The overall design of the interaction and sound producing architecture was based initially on structures proposed by Eacott and d'Inverno [5, 4] and subsequently developed by the Ambigence team. Its main elements are summarised here.

In order to offer music which is *deeply* interactive, and that would offer genuine scope for new and personalised music which reflected the tastes of the users of the space, the sound would have be generated in real time by the use of *algorithmic structures*. Algorithmic music can be manipulated very explicitly. For example, reducing the tempo by 10% or transposing the piece up a tone can be dome much more effectively with algorithm representations of music rather than sampled ones. More sophisticated manipulations included changing the basic underlying scale (from major to minor) and the reharmonization of a section or melody. However, the street does make use of some pre-recorded *sample*elements but it should be noted that these are much less manipulable. In cases that samples were used we would aim to use algorithms which adapt the sample, i.e. by cutting or adjusting in some way rather than hearing them 'straight'.

2.2 User Input

The commands that are recognised by IStreet are almost totally arbitrary and are predefined by us. In future we would anticipate a language being developed by users organically, and that this language became the mechanism where users could communicate with each other and with our interactive sound installations about how to *build* the journey of a musical experience. One reasonably straightforward method would be for users to save a set of commands (such as *warp* and *dark*) as another command (for example, *nightingale*) so that IStreet would subsequently recognise *nightingale* as a new command either from that used alone or from some user group.

It is worth noting that whilst some of these commands are quite explicit some are much more ambiguous in their possible impact on performance. We were interested in how such difference affected users expectation and satisfaction. These commands were as follows.

`8bit, air, cheese, chill, complex, dance, dark, elevate, energise, frenzy, meditate, mellow,`

`minimal, new, refresh, space, spicy, tango, urban, warp, xtatic`

As we have said above, we are interested in building autonomous interactions where the language of interaction would be organic, dynamic and fluid.

One of these text messages from a user is sent to a number supplied by Vodafone and they appear on the local installation as follows.

`urban|070143000|2003-10-22 18:34:18<BR>`

`frenzy|0707143000|2003-10-22 18:34:16<BR>`

`dance|0707143000|2003-10-22 18:24:15<BR>`

`tango|0705580000|2003-10-21 22:30:22<BR>`

`urban|0705580000|2003-10-21 10:29:50<BR>`

As soon as an SMS reaches Vodafone it is inserted in a new text document (formatted according to our specifications with | as a separator and
 at the ending) located on their server. The software program *SMS decode*, written in Python, reads this document once every second and searches the top 4 rows for new commands. In this way the maximum number of commands the system can actually process is four per second. SMS decode performs simple checks like spell checking and filtering. When a command finally is confirmed as being a legitimate system command it is sent as input to SuperCollider that puts it in line. In summary, when SMS decode finds new commands it will look at the prefix and filter out any country heading such as *uk* or *se*. It will then convert all characters to lower-case, check if the command is in the dictionary (spell checking), check if the command is allowed and finally pass it on to (command+phone number+time) to SuperCollider.

For example, in the case recent new messages would be stored ready for processing.

istreetbuffer = [urban, frenzy, dance]

These queued messages are simultaneously processed by the SuperCollider program at specific pre-defined times. In the present implementation, all the music is beat-based, structured into bars and in 4-4 time. Accordingly we choose to interpret these messages every 16 bars. Once the messages have been processed the buffer is cleared.

2.3 IStreet Data Structures

We now introduce a description of the main data structures that we can used to describe the software architecture of IStreet.

2.3.1 Style

The starting point for all sound would derive from a set of pre-composed compositions which we referred to as *styles* as they were an attempt to capture some essence of a musical genre such as, for example, *funk*. These styles took the form of compositions written within a *common style framework*, which we will consider in more detail later. It was absolutely necessary to have a common framework for writing pieces (styles) for this piece so that it would be possible to make intelligent, musically-sensitive operations at playback, especially when it came to combining them for performance. In our piece the styles were written algorithmically by Eacott, Lörstad and Olofsson. In future incantations we envisage that styles would be developed by users of the space in which this application was situated.

2.3.2 Tracks

A track can be thought of in the very traditional sense of a track in a recording studio or as in standard sequencing. In the current IStreet version each style has exactly ten tracks. Moreover, we have pre-specified the role and function of each of these tracks and given them names as follows. We also associated each track with a *level*. The idea is that the lower the level the more "key" or "defining" to the overall sound that particular track is.

This information can be found in the following table.

Level	Track	Name	Role
5	1	Percussion	specific genre effect
4	2	Samplebeat	sampled sound
3	3	Bass drum	typically every beat
2	4	Snare	off beat, patterns, fills
1	5	High hat	key in defining style
1	6	Bass	bass riff
3	8	Pad	ambient backing
2	7	Chord	harmonic sequence
4	9	Melody 2	generic melody
5	10	Melody 1	style related melody

In order to create the framework we had to make some general assumptions about its likely content and the way it may be permitted to change. We agreed on a model which used 10 instrument channels, where each channel is on a different level. Also notice that the non-melodic (called rhythmic or percussive tracks) tracks are from tracks 1 to 5 and that the melodic instruments range from 6 to 10. Therefore, each level has a percussive and a melodic track component.

2.3.3 Instruments

Naturally, by limiting the number of instruments this way we are putting some restrictions on the types of sound possible. This simplification was necessary however in order to offer a simply understood ubiquitous framework to the composers and maintain a manageable level of complexity in the software design. Composers were allowed to define their own instruments to be called by any of the 10 channels. Although there was no absolute restriction on the scope of that instrument it was suggested that the sound it produced should relate to the instrument channel. For example, a drum sound should not be used in the *Melody1* channel as its behaviour would be inconsistent. Instrument channels themselves can consist of more than one 'layer' of that instrument (for example there are typically many layers of hi hats on the *HiHat* channel) and composers need not use all 10 channels. Composers would also be free to choose from a growing 'library' of existing instruments and effects.

2.3.4 Structure

Musical material relating to each instrument channel would take the form of 'patterns' (using the Patterns class library in SC2). The composer could build a structured arrangement of each style by arranging the patterns into 'sections' such as intro, verse, chorus, bridge etc. Conventions were also agreed about the organisation of melodic and harmonic structures so that these elements could be read and adapted by any style.

2.4 System Development

The actual output of music would derive from a combination of these styles in some way (bhangra and ambient for example) in and a range of interaction parameters which affected a wide range of attributes from tempo to instrumentation and other sound quality aspects. We used the notion of 'current position' of the system to specify its state at any time. The current position would be a position defined by a co-ordinate in the multidimensional parameter space. For simplicity, we would only consider the 3 styles nearest (on the style parameter plane) to the current position.

IStreet prototype version 1 was based on this model. It produced musically interesting and attractive results but was dangerously high in processing requirements, as it had to analyse and combine 30 instrument channels (3 styles each with 10 instruments) in real time.

For the 2nd version we decided to use a maximum of 10 instruments (we used the metaphor that the IStreet 'band' should consist of 10 musicians rather than 30) and that the part each 'band member' played would be created by combining (or 'forking') elements of each of the 3 current styles.

After a long process of re-design it was discovered that although it worked, this method produced results that simply were not musical!

For the 3rd version and in the face of a looming production deadline we opted for a very simple way of fusing styles by simply replacing whole instrument channels two at a time. I.e. if we begin with 100% funk all the instrument channels are funk. As we move towards bhangra the inner 2 channels (5 percussion and 6 bass) were replaced by the bhangra instrument channels, as we move to 40% bhangra the next two channels (4 and 7) would be replaced. In other words, this is a simple first-in first out-buffer with 5 positions (10 tracks grouped in pairs). When a new style is called it will push out the 5th of the 5 currently playing styles. This method worked simply and effectively. An added benefit was that rather than combining elements of just 3 styles up to 5 could now be combined. The musical results seemed sufficiently good too.

3. CREATING MUSIC AS STYLES WITHIN A UBIQUITOUS FRAMEWORK

3.1 Compositional Structure

For the installation to make intelligent (and musical) decisions about performance composers are also instructed to use a specified *form*. Essentially each style consists of four *sections*, each section consists of sixteen (16) bars, and each bar consists of 4 beats. The order in which these sections are played is currently predefined in a global variable. One way to intuitively think of sections is as follows (though we do not make that a requirement in the current implementation).

section 1: Intro
section 2: Verse
section 3: Chorus
section 4: Bridge

These parameters effect the basic sound as defined by the styles. We can have as many global variables as we like in theory but in practice we have a handful such as tempo, key and so on. In addition we have some global effect parameters such as reverb, delay, pitch effects (high and low), eight bit (sampling rate), phaser effect and warp effect(amount of cutting of the snare drum).

Each of these variables has a predefined system default value.

3.2 Blend

A blend is a piece of music made from different tracks of the library or existing composed styles. Here is an example of a blend.

Level	Track	Name	Style	Muted
5	1	Percussion	Funk	Yes
4	2	Samplebeat	Tango	Yes
3	3	Bass drum	Funk	No
2	4	Snare Drum	Trance	No
1	5	High-hat	Funk	No
1	6	Bass	Funk	No
2	7	Chord	Trance	No
3	8	Pad	Funk	No
4	9	Melody2	Tango	Yes
5	10	Melody1	Funk	Yes

The blend, the muting of certain tracks, the value of the global variables and the current section uniquely determine what music is output.

4. DESIGNING USER INTERACTION

The commands chosen to feed to the system and alter the music with are specifically thought out to be "non-musical" terms. A command such as "cheese" or "air" gives the user the option of freely associating to what he or she means by such a word rather than building up the expectations a musical term would. However, there are some universal meanings to some terms which the Ambigence group have considered and interpreted. *Dark*, for instance, emphasizes the lower frequencies of the music, cutting the higher frequencies and also diminishing focus on the tempo to give the music a *darker* timbral quality. When commands such as *dark* are repeated the music is affected increasingly, until an absolute lower bound is reached that was pre-defined.

In all there are twenty commands available that all have different functions. Eight of them correspond to *styles* composed in a certain genre and their use adds elements from these styles to the soundscape.

Name	Style
Cheese	funk
Spicy	bhangra
Xtatic	trance
Dance	techno
Tango	tango
Meditate	ambient
Frenzy	drum'n'base
Urban	hip hop

The styles represent a composers attempt to capture the essence of a musical genre, similar to the notion of *signatures* developed by Cope [2]. The styles we chose were based on our perceptions of what students currently listen to.

As we discussed earlier, the delays of sending SMS messages meant that commands could never effect the musical output instantaneously, hence they were queued for action. It is also worth noting that not only did the user get the feedback from the display, and the change in musical performance, they also received a message from Intelligent Street thanking them for taking part in this project. This proved highly motivational, as many of the participants were encouraged by this explicit text-based acknowledgment.

Choosing one of the above commands adds an element of the corresponding song by adding a track to the overall soundscape. For every time the command is sent to the system one track of the song is added.

The remaining thirteen commands have functions that affect the music in different ways:

- *air* emphasizes hi parts, adds reverb, reduces complexity and removes samplebeat
- *mellow* reduces beats, adds a low pass filter and emphasizes the mid section
- *minimal* reduces data in melody patterns and reduces tonal content
- *chill* removes all rhythm elements, adds a low pass filter and a delay, emphasizes on long durations
- *refresh* randomizes styles, removes style processes, normalize
- *new* moves to random style
- *dark* removes higher parts, applies low pass filter to rhythm tracks
- *complex* makes tonal and rhythmical parts more complex, applies random pitch filter, scrambles and creates new rhythms
- *energise* raises tempo, make rhythm part more complex, raise pitch
- *elevate* raises tempo, emphasizes hi parts, adds a phaser effect
- *space* reduces data content, removes melody and rhythm part from top down, adds a delay ambience
- *8bit* applies 8bit degrade
- *warp* uses warp cutter

For every time one of the above commands is received the system alters the sound accordingly. All the commands except for *new* and *refresh* follow the same procedure as described above and have a more or less subtle effect on the music, whereas the remaining two result in a more drastic change.

This information might have given a better insight of the system for the user and possibly would make the user explore more of the available combinations.

Some tracks may be muted (this is like turning the volume off on a particular track). The overall density of a piece is the percentage of un-muted tracks. In the table above the current density is 60% as 4 tracks are muted. The style density of a blend is simply the percentage of tracks that come from a particular style. In the case above, funk density is 60%. The *dominant style* is the style with the highest density.

4.1 PlayBack

The first command starts the system and the command chosen has a preset combination that corresponds to to a certain setup. For example starting from idle by sending the command *spicy* will play a mixture of 40% bhangra, 30% urban and with a density of 70%. In general, all ten tracks of that system are played with certain default values for global effects like delay and tempo.

If no more commands are input then the music will decay over a period of around a minute. The system reverts to outputting a soft, barely audible background until the next command is processed. Every 16 bars all queued commands are processed by IStreet simultaneously and the new master blend is output.

4.2 Commands and Methods

In the current system there is a one to one mapping between text commands and internal IStreet *commands*. Each IStreet command then calls a predefined set of *methods*. Whereas commands relates to a desired effect at the user level, methods are lower-level operations on the IStreet data structures. Some example methods are listed below.

- *increase density* un-mute the highest level muted tracks.
- *decrease tempo* reduces the beats per second by a given amount
- *transpose tracks* transposes melodic tracks according to some array of order and amount
- *lowpass* applies low pass filter to rhythm tracks
- *melodic filter* progressively reduces the set of allowable notes

As we discover more about user expectation, we can change the particular set of methods that are invoked by a command. Our intention was never to be too prescriptive about what commands should or should not do. In an ideal autonomous system, that's for user groups to determined over time. Our initial attempt at defining the complex command is as follows: *increase density, reduce active lowpass filters, decrease melodic filter, transpose random tracks.* Again in general, we anticipate that the relationship between user commands and the methods it calls be a function of usage in future versions.

5. RESULTS OF EVALUATION

Evaluation has been performed by Umeå Behaviouristic Department at the School of Music in Piteå. 41 subjects were categorised by gender, age, education, and whether they were part of the media department or the music department.

The subjects were asked to interact with IStreet for a short amount of time (5 to 10 minutes) and after that kindly asked to answer a number of questions written in a questionnaire. 20 questions were formulated to try and find out whether the interaction was sensed as comprehensible, and if other areas could be imagined where a framework like this might be used. The subjects were also prompted to answer if they were likely to use mobile phones to control their environment in general and applied to the musical parameters of Intelligent street in particular.

The group of subjects aged from 19 to 40 years, the majority being around 25, and these formed the selection from which this evaluation has been made. A vast majority (95%) of the subjects claimed to have a great interest in music, nearly 80% played an instrument. The relationship male-female was around fifty-fifty.

Questions regarded matters such as:

- Is this a comprehensible way of affecting sonic environments?
- Does it lead to an improvement of the surrounding soundscape?
- Would a system like this be desirable in other spaces?
- Is the mobile phone an appropriate tool for similar usages?

Results were quite varied. The sense of interaction and that they were indeed affecting the soundscape was felt by 98% of the test group, a figure that is very satisfactory as this was one of the most central goals with the whole project. A majority (70%) of the media students but only half (52,5%) of the music students felt that they appreciated the change that they made on the soundscape. One possible explanation to this might be that media students are more open to to using new technologies for communication and conveying emotions than music students. Another explanation to this could be that more effort should be put into composing the music, a critique that the composers of the Ambigence group would have to consider.

About half of the people interviewed could see other areas where they would want to be able to control the sonic environment. There was a considerable difference between music and media students, 70% of the latter had an interest in changing the sonic environments, whereas corresponding figure for the music students was 38%. 20% stated that they had no interest in changing the sonic environments at all. As to in what places people were inclined to modify the surroundings answers showed a difference in both locations and gender. The restaurant/pub/disco surroundings was found to be the environment divided the subject group mostly from the gender aspect. 60% of the men wanted to modify these surroundings whereas women were much less inclined (15%) to modify the soundscape in such a place.

64% stated that they thought that the mobile phone was a suitable tool for modifying and improving surroundings but 57% were reluctant to make use of these services if they had to pay SMS-rates for it. To the question if they found the mobile phone a time consuming tool to use 70% answered no.

Another curious fact is that approximately 35% of the male part of the subjects found the system difficult to understand. This problem had no representation among the female part.

Suggestions to optional places where IStreet could be set up ranged from creative ideas such as "...public places,shopping malls, restaurants, waiting areas..." to less enthusiastic ones like "...nowhere, I want the environment to be quieter".

6. CONCLUSIONS

The Ambigence idea of having commands that didn't refer to the music in technical terms turned, according to this evaluation, out to be something that provided user satisfaction. By not using words directly related to musical performance (eg using *energise* rather than *transpose*), we were able to instill a sense of curiosity as well as expectation in users. Evaluation, however, did take place at a higher-education institution, which accounts for the great interest in music and the high rate of practicing musicians. Though in general, there was clear evidence that such an installation would be welcome in more general environments. It would be interesting to compare this evaluation to with one performed in a totally different context, such as a shopping mall or maybe in connection with a sports event.

Questions of coherence and a certain predictability as opposed to subtlety and randomness have been discussed within the project group. The changes made to the music can at times be rather subtle which some of us thought would give a confusing impression and make the user feel that the interaction does not affect the course of music at all. However, this turned out not to be the case, as the above figure of 98% clearly showed. Possibly if the changes were made even more subtle maybe the students of music would have a higher percentage of users appreciating the musical structures more.

The main emphasis of the project lies on on the human computer interface and their interaction. To this we add the mobile phone and through text messages we influence our sound environment as well as being the link of communication to the outside world. By sending these text messages

users will be affecting your immediate surroundings as well as communicating and sharing experiences with people in other locations. Our ambition is to set up a network of Intelligent Streets in other locations thus connecting London and Piteå to a global network. We wish to explore ways of inviting people to participate in creating interactive sound art and find new ways of utilizing mobile phone technology as an art interface.

6.1 Concluding Remarks

In years to come we would like to build an installation that could function autonomously [9] without the need for any outside assistance from programmers or musicians. Ideally, we would build a system that could learn how best to respond to the combined inputs of user groups in a specific location over time. Moreover, we wish to explore the possibility of users collectively and interactively *composing* music, using techniques from intelligent agent systems [10] coupled with *generative music*, where random elements in the algorithmic structures generate unique performances.

One consequence of recent technological advances is that we can imbue everyday objects with processing power. Not only will this enable these objects to make decisions about what to do, it will also allow them to communicate with each other and make sophisticated decisions. This leads to the notion of Ambient Intelligence where computational entities are interwoven into the fabric of our lives.

We envisage a scenario where music devices can not only make intelligent, sympathetic decisions about sound generation in order to satiate the particular requirements of a user, but where they would also interact with other devices both musical and otherwise in a massively dynamic and heterogeneous environment. We envisage that there will be a shift in the pattern of behaviour relating to music consumption; a move from the passive consumer to the active creator.

Shared environments should be designed to take advantage of recent technologies, and artifacts developed that encourages social interaction between the people in it. It is our belief that in future users should be encouraged to interact and negotiate about all manner of environmental issues such as heat, light, humidity and so on. Our work has been an investigation into exploring the fundamental social notion that it is the collective, dynamic interaction of users, that should determine the environment in which they want to be. This is an initial prototype exploring the potential of some very novel ideas in environmental installations using new techniques from ambient intelligence, but the future looks (and sounds) bright.

7. ACKNOWLEDGMENTS

Fergus Rougier is a sound designer specialising in functional music and sound and has been part of the Ambigence team through the development of this installation. We would like to thank The University of Westminster, especially the Harrow and Cavendish campuses, and their respective provosts, Professor Phillips and Dr. Tyler. The Interactive Institute in Malmö and Piteå, and Acusticum, particularly Nina Sjömark. We are indebted to Jan Berg and Anders-Petter Andersson who gave helpful comments on this paper. We would like to thank the evaluators of IStreet: Roger Arvekev, Andreas Doverhäll, Adam Sundström and Johan Wirmark. Finally we would like to thank Vodafone UK and Vodafone Sweden who have supported us throughout his project and whose assistance we gratefully acknowledge, especially Andrew Bowen, Patrik Holst and Göran Brandt.

More information on this project can be found at the addresses below.

`www.intelligentstreet.net`

`www.tii.se/sonic/intelligentstreet/`

8. ADDITIONAL AUTHORS

Additional Author: Fredrik Olofsson, University of Westminster, `f@fredrikolofsson.com`

9. REFERENCES

[1] A. Andersson, B. Cappelen, and F. Olofsson. *Unfoldings*. Interactive Institute in Malmö, Sweden, 2004.

[2] D. Cope. *Virtual Music: Computer Synthesis of Musical Style*. MIT Press, 2001.

[3] M. d'Inverno and M. Luck. *Understanding Agent Systems (Second Edition)*. Springer, 2004.

[4] J. Eacott. Smartsound: a framework for multi-user sound interaction in R. Barbrook, J. Eacott and D. Jennings (eds). *Proceedings of the CREAM Symposium, Cybersonica*, 2003.

[5] J. Eacott and M. d'Inverno. Embedded intelligent music or ihifi the intelligent hifi. *Digital Creativity*, 14(2), February 2003.

[6] M. Eastley. *Bamboo circles 1991*. Exhibited at the *strangeAttraction* exhibition, Morley Gallery, London, 1999.

[7] J. Finer. *Longplayer*. ArtAngel, London, `www.longplayer.org`, 2000.

[8] M. Luck, R. Ashri, and M. d'Inverno. *Agent-Based Software Development*. Artech House, 2004.

[9] M. Luck and M. d'Inverno. A conceptual framework for agent definition and development. *The Computer Journal*, 44(1):1–20, 2001.

[10] M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing*. AgentLink, 2003.

[11] J. McCartney. *SuperCollider audio synthesis environment*. `http://www.audiosynth.com`, 2004.

[12] SSEYO/Tao. *http://www.sseyo.com/news/pr20030910.html*. 2004.

[13] D. Toop. *Ocean of Sound*. Serpents Tail, 1995.

[14] D. Zicarelli. *Max/MSP*. Cycling 74/Ircam, 1990-2001.