GLITCH LICH: EVOLUTION OF AN INTERCONTINENTAL NETWORK BAND

Chad McKinney
University of Sussex
Department of Informatics

Curtis McKinney
University of Bournemouth
Creative Technology Group

Ben O'Brien
University of Florida
Department of Music

Cole Ingraham
University of Colorado at Boulder
Department of Music

ABSTRACT

In this paper the authors, as members of the laptop network band Glitch Lich, discuss their evolution over nearly five years of performance. We detail our transition from a group exploring game theory to performers of network music and programmers of networking software. A brief overview of the technology developed and used by the band is provided. In conclusion, we present a collection of new directions for the ensemble and network music as a whole.

1. INTRODUCTION

Glitch Lich is a laptop band comprised of members Cole Ingraham, Chad McKinney, Curtis McKinney, and Ben O'Brien. This quartet is unique in that its member reside in three different time zones, yet still perform together despite separation by thousands of kilometers. To cope with over seven and a half thousand kilometers of separation the quartet has had to devise innovative methods to continue collaboration.

The group met at Mills College where their studies focused on the practices of improvisation with electronics, just-intonation, and game theory. Early works utilized Open Sound Control (OSC) based networking, however public performances of these compositions were plagued with packet loss, dropouts, or disconnections. Glitch Lich resolved these network problems with the creation of OSCthulhu, a flexible system of abstracted *SyncObjects* and *SyncArgs*. Inspired by network mechanics found in multiplayer videogames, members of the band formulated the open-source client-server architecture to allow for better synchronization of audio and visuals for global performance. OSCthulhu is now at the core of every Glitch Lich performance, allowing the ensemble to perform on slow and even intermittent connections.

Network music compositions first emerged in the works by the League of Automatic Music Composers. Each new composition was a product of the soft/hard-ware instrument configurations designed by The League members. Communication between members was achieved by directly soldering connections between computers, which subsequently resulted in a fragile network prone to errors. Years later, the computer band the Hub was created, including former "League" members, and utilized a server based approach to networking. The system allowed members to share a common memory for the ensemble as well as communicate via a client-server architecture. Hub member Scot Gresham-Lancaster writes,

> "... [T]his music is idiomatic and requires a special understanding of the software and hardware being used. The inherent ironies in this approach are clear: these pieces of music can never be repeated exactly, and it would be very difficult to reconstitute the exact state and setup that made up a given performance." [6]

With the emergence of MIDI, the Hub adapted a MIDI-based system which simplified the network upkeep but absolved a shared memory environment which, in turn, altered their musical approach. John Bischoff, a member of both The League and the Hub, recalls,

> "...the relationship between issues of music time vs. hack time and robustness were not very different in the last analysis. I see both groups as having a slightly different angle on the concept of network music but at the same time being equally successful musically and technically."[3]

The work The League and the Hub created demonstrates how their specific artistic demands were both satisfied and altered as a result of the evolution of computer technologies.

Over the past decade there has been a significant presence of network laptop orchestras around international universities. With the advent of the OSC protocol, countless university laptop ensembles began surfacing and composing works which utilize User Datagram Protocol (UDP) messaging [13] [11]. Audio programming languages such as ChucK and SuperCollider have implemented OSC, allowing for bands such as PowerBooks Unplugged to share code in real time [9]. There are many benefits to this form of communication, mainly that group members may be remote and it is not necessary to have their individual public and private IP endpoints. Additionally, the strain on the server is minimal as it is only a point of transmission between users.

The Princeton Laptop Orchestra (PLOrk) is an example of a laptop ensemble which has written works centered around OSC communication. For example, in the composition *CliX*,

> "a machine conductor synchronizes and quantizes the sounds triggered by each players keyboard by emitting periodic pulses via OSC, leveraging the computer to augment the degree of control offered by the keyboard."[5]

Though OSC is straightforward and robust, it does have several shortfalls. The use of UDP requires that a system rely on extensive synchronization between peers which may suffer breakdowns due to packet loss.[1] With this knowledge, network music ensembles must coordinate appropriately to prepare for the potential problems of lost data.

This article explores the aesthetic and technical approaches of the laptop network band Glitch Lich. A brief series of statements from the members is followed by a historical documentation of their performance practice. Afterwards, the penultimate section addresses technical details, before a conclusion including comments on potential future work.

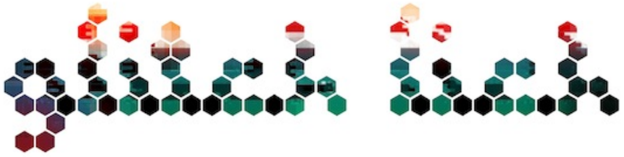

Figure 1. *Glitch Lich logo*

2. AESTHETICS

Networking is not a style or genre, but rather a technique. As such, there can be many ways to write music for such an ensemble. The following are a series of statements by the members regarding their approach to writing music for a network band. By including statements from each member, the differences and similarities within the group will be highlighted, providing a holistic representation of the group.

2.1. Chad McKinney

Glitch Lich is a band with a philosophy and any philosophy deserves a manifesto[12]. Glitch Lich demands: 1. Steal. Steal from the performers, from the audience, from the internet. What is out there is for the taking. 2. Share. Make what is yours available to the performers, the audience, and the internet. Don't hide behind your computer. Also, if you have to chat, show the messages. 3. Don't be used, use. Performers are composers, there are no second class citizens. Every member must write music and code and do so often. Make yourself better and push each other. 4. Network music is not about tools, but relationships. Maintain control over all the elements of a performance, but keep the focus on the band.

2.2. Ben O'Brien

Performing with Glitch Lich is unlike most ensembles. Glitch Lich composers still have rigorous demands for the other members, and there is a certain need for work-shopping. But these necessities are allocated to atypical-to-traditional-music-performance directions. A commonality is an understanding of all the facets of the performer's instrument; in this case: the computer. In short, knowing the whereabouts of files, ability to execute files in real-time, network accessibility, etc., can be compared to knowledge of fingerings, dynamic control, extended techniques on the guitar. This of course is basic knowledge, and the best way to better understand one instrument is to push it into arenas unfamiliar to one's own knowledge. I am reminded by Paul Lansky's quote:

> "If you want to work at a professional level, I think it is best to think of yourself as somewhat analogous to a professional driver. You ultimately want to design your car, and when it breaks you want to be able to pull over to the side of the road, lift the hood and fix it yourself."[4]

Members of Glitch Lich have inspired me to better understand my instrument. Composing works for them have revealed worlds of problems that I would not otherwise uncover, and as a result I have discovered personal compositional interests relating to composition, improvisation, and computer translation.

Speaking to the aesthetics of Glich Lich, I first look to our method of networking. As Glitch Lich employs a network system for communication which resembles that of online multi-player games, it is logical that the music reflects this type of communication. While there are some sonic qualities similar to popular 8-bit video games from the 1980, Glitch Lich's approach draws from the vein of Classical music: present a musical idea, elaborate on it, propose the question - to the listener - of whether the musical state is sonically similar to the original, and then decide to return

[1] Packet loss occurs when one or more packets of data are sent from one network endpoint and fail to reach their intended destination.

to the initial idea, or repeat the process with a new musical germ. In works such as *Neuromedusae* and *Yig*, there is a score that dictates the cycling through SynthDefinitions, allowing performers and listeners alike to draw similarities to the materials. The result is a sense of forward-trajectory while relying on a relative process for music-making. Personally when composing for Glitch Lich, my focus is shifted to designing structures that concentrate on interactivity, malleability, and noise - which is not to say that my personal work does not engage in this manner. But rather, my concern lies that given my knowledge of the performers, how will they push my structures into unknown realms, and how will their collective mindsets effect the composition. With every performance there is definitely a sense of excitement which I believe is incredible given all the performers are scattered across the globe.

2.3. Curtis McKinney

Network music is something of a personal matter for me. It found me, but in a lot ways it's perhaps the kind of music I feel most comfortable with. Growing up with a twin brother(Chad McKinney), collaboration has been my natural mode for making music my entire life. Thus, when Chris Brown and John Bischoff introduced us to the concept at Mills College, it simply made sense. Why *not* play electronic music with the same fundamental collaborative energy that we have been using our entire lives? However, for me the promise of network music is more than that. Not only may you play with the same level of interaction as an ensemble of acoustic instruments, network music allows you to reach down to an even deeper level of interaction and dynamism.

To me that is the heart of the matter, and that is exactly what Glitch Lich as a band stands for. While being able to perform with my bandmates despite our different locales has been extremely useful, it is not really the point of the music. In fact it would actually be more advantageous if we were to be in a single locations. For me, the central theme is to construct an assimilated whole that is greater than the sum of its parts, producing a music that is the combined efforts of its participants. During this interplay, there is a constant shifting of roles, interaction, dependency, and identity, it brings an exciting liveness to performance that satisfy all my musical desires as a performer. This oscillation between group-identity and self-identity is part of the narrative drama of performing with Glitch Lich. In a rather fluid manner your performative gestures may slip from solo, to accompaniment, to joint manipulation of a single sound, to losing the ability to exactly decipher just how your action is manipulating the group's electronic morass.

I've attempted to give a label to this style of music making that I'm trying to achieve with Glitch Lich. Given that these electronic compositions are essentially serving as artificial instruments played by multiple individuals, I have come to call them Network Based Multi-user Instruments, or NMI's. The aesthetic construction of these NMI's has been the central focus of my research during my PhD studies. For an NMI to be successful in the setting of Glitch Lich I feel that there are certain criteria I want it to achieve. The instrument must allow for close-knit real-time interactivity of performers, it must have the ability to be composed for, and it must have some sort aestheticized visual presentation that helps to clarify the music and the network interactions to the audience as it is performed. The visual presentation of these instruments has become a more and more important facet as the band has evolved. What I've discovered is that while people have a strong willingness to understand and appreciate the underlying structures of network music, sometimes it is rather opaque. However it doesn't have to be this way, and the more I've worked to let the audience in past the liquid-crystal wall, the more they seem to appreciate exactly what's going on, and thus appreciate more the music itself.

2.4. Cole Ingraham

The ability to collaborate with musicians in real time over any distance in any situation where there is internet has opened up an amazing amount of possibilities for me both as an artist and performer. Being a part of any long standing ensemble always brings with it a high level of musicianship which every performer appreciates. The simple fact that the members of Glitch Lich were moving to different parts of the world would mean the end to most groups. However, due to the nature of network music this is a non-issue.

In fact, our geographical dispersion has been beneficial for us. It is easy, and rather common, for Glitch Lich to perform in the UK one night, and the US the next. We have even occasionally performed in multiple locations simultaneously. Being able to tour in multiple states and countries without needing four plane tickets is something that most quartets could only dream about.

This mode of working also inherently fosters collaboration: not only in performance but in the creation of the music. It is commonplace for one member to initiate a piece and the others constantly contribute to its development. The entire group then feels a sense of ownership in every work and that makes every performance that much more personal.

3. PERFORMANCE PRACTICE

Performing as a laptop ensemble requires many decisions to be made about the interactions between the performance and their presentation to the audience. When Glitch Lich (then known as the League of Art Game Composers, or LAG) began performing together in December 2007 the ensemble was looking for ways to make their laptop performances more relevant to the music and the audience. Previous work from the members went little beyond tweaking GUI knobs as a fundamental performance element. Audiences were confronted by the back of laptop screens and the opaque information provided by occasional typing as cues for the performer's involvement. The history of the group is as a series of solutions where each new system is some attempt to solve the problem in a new and novel way.

3.1. Game Systems

The group's first attempt to solve these performance issues was to use games and game like systems as fundamental structures. This approach was inspired by several composers and artists including John Cage, Marcel Duchamp, Yoko Ono, and John Zorn who incorporated game structures into their work [7]. By using boards games, card games and video games, the member's actions during a performance became not only relevant but more transparent to the audience. *Red King Snoring vs The Octopus Knight*, a work featuring an electronic chess board connected to a laptop running SuperCollider, sonified the movement and capture of pieces allowing the natural structure of the match to govern the flow of the music. By using the game as the input into the system, performer actions were made more transparent, however the were still many layers of abstraction in the system. The mapping of any event is arbitrary when it could be mapped in any number of ways, so while the performances became more theatric, the level of direct control over the music decreased.

Several works by members explored the use of games and the varying ways they could be incorporated into a performance. Often the game itself was a controller, such as the work *Castlevanian Noise Vampire* by Chad McKinney, where the performer plays a gameboy and the output is manipulated in realtime. However other works, such as *Pong* by Cole Ingraham, flipped the problem on it's head by using pitch and onset detection of instrumentalists to control a video game. From 2007 to 2009 the group used several games such as Chess, Samurai Showdown, Koi Koi, Pong, Castlevania, and invented games to control laptop performances. However after almost two years of experimentation the group wanted a new direction.

3.2. Local Networks

While studying with John Bischoff and Chris Brown at Mills College the members were encouraged to make network music. The ensemble's previous work only indirectly addressed performer interactions and often times performances only used a single laptop. The group decided to change direction and attempt something more similar to the League of Automatic Music Composers or the Hub. New works were written and new systems were made where members used laptops connected to a local router via ethernet to facilitate network performances. Concerts featured the ensemble within a sprawl of cables and interfaces on two large, and often plastic, tables. By now the group had solidified itself as a laptop quartet with members Cole Ingraham, Chad McKinney, Curtis McKinney, and Benjamin O'Brien. Coding skills became increasingly important as each new system pushed the complexity of the group's codebase.

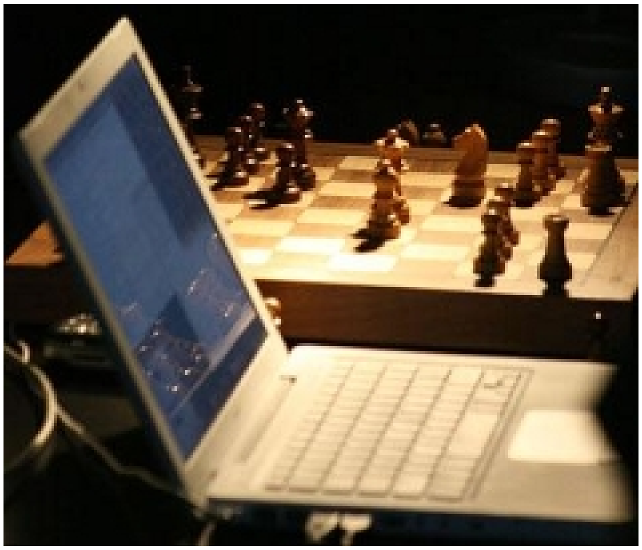

Figure 2. *Performance at the 2008 Signal Flow festival in Oakland, CA*

The concert at the University of the Pacific in November, 2010 is a typical example. The group performed several works featuring custom systems that enabled rhythm splicing, code sharing and manipulation, and game elements. The theatrical interfaces such as the Chessboard and Koi Koi cards were missing. The ensemble became more computer centric and because of this took on the problems of that presentation. However now that the ensemble focused on network performance, each member's actions were important to the work at the most fundamental level. Portraying these connections to the audience was still a challenge and often visualizing programs were designed to break open the systems for the audience's benefit. The network connectivity also introduced the ability for the members to use chat during the performance, allowing fast and silent communication that was only possible in a computer ensemble.

The group was performing regularly, however organizing rehearsal could be difficult. By using OSC for networking they were able to rehearse in separate locations, however their systems were not designed to withstand packet loss which could create asymmetrical program states quickly. Furthermore, the group's reliance on OSCgroups, a system for routing OSC messages between users, introduced significant problems due to firewalls and routers [2]. A member's router model or current location (often colleges) could completely block osc traffic, preventing rehearsal or performance.

The group also experienced the growing pains of maintaining an increasingly large and changing codebase. Their reliance on SuperCollider required each member to keep copies of other member's classes on their system, however this introduced many problems when members were on different versions of SuperCollider. The practice at the time of using Google Wave or e-mail to disseminate and track code updates created problematic scenarios. Rehearsal were marred by bug hunts and often no piece was even completely played through.

3.3. Distributed Networks and Current Practice

February 7th, 2010 marked the first multiple site performance by the ensemble. The concept was ambitious: Cole Ingraham was present at Montana State University while Chad McKinney and Benjamin O'Brien performed from Mills College in Oakland, CA and Curtis McKinney in Oklahoma City. The concert featured new pieces with advanced user interfaces written in Java to communicate with SuperCollider. These new systems featured locally rendered audio on all machines allowing for performers to be in different locations without streaming audio. It was called "Digital Sunrise of the New Era". Despite the grandios title, the performance with rife with technical problems.

The ensemble was able to find time to rehearse before the members separated, however this was always using the same router. At the performance the firewalls between Mills College and Montana State University proved to be too difficult for OSCgroups. Throughout the course of the concert there was realization that the members in California could communicate with Curtis in Oklahoma, and Cole in Montana could get some of the communication from Oklahoma, however California and Montana were completely separated. This created bizarre system states and furthermore the performers had no way of communicating to each other inside their systems.

After the show the group decided some measures needed to be taken to solve many of the problems with the ensemble. Version control through Subversion became mandatory for any code that would be shared among members. This quickly cleaned up the code base and made updates seamless, especially for projects with multiple authors. To solve the problems with OSCgroups, a new OSC client and server architecture was proposed. Something that could deal with any firewall or router and would also feature synchronization to overcome packet loss. This led to the development of OSCthulhu which will be discussed in further detail in the next section.

Glitch Lich has performed many times since the Digital Sunrise event, and has experiences several other issues in subsequent performances. Before a recent performance in Boulder, Colorado, band members were still attempting to compile key software components after an update. By the time of the concert, only two members were able to perform. It has since become a rule that authors of software must provide precompiled binaries for both 32bit and 64bit architectures and for all versions of Operating Systems that are used by the various members of band. Furthermore, every system must have a fail-safe mode that allows it to seamlessly transition from networked to local modes and back, for when network connectivity is completely lost or becomes too unstable. This rule follows the group's performance at the 2010 SuperCollider Symposium in Berlin. During their performance of the piece *LAGMonster* the server crashed. Without a local mode, the piece was halted after only thirty seconds.

These measures have proven to be quite effective and Glitch Lich has gone on to perform regularly as a transcontinental ensemble with members in three different time zones. In fact it has been over two years since all four members have been in the same location. However because of the separated locations of the ensemble, performances now must be concerned with how to portray members in separate locations to the audience as well as guide performances without the use of visual cues. Rehearsing can be difficult as well with a seven hour difference between the furthest members of the ensemble. However, Glitch Lich continues to research new possibilities for network music and maintains a steady performance schedule.

Figure 3. *Performance at the 2012 Network Music Festival in Birmingham, UK.*

4. TECHNOLOGICAL FRAMEWORK

Throughout its history as a band Glitch Lich has gone through several iterations of technological frameworks to facilitate network performances. As the sophistication of the group has increased, and as the aesthetics of the ensemble have changed, several new technologies have been developed and utilized to match the requirements of the band at that time.

4.1. OscGroups

For the first iteration of Glitch Lich's networking technologies the NAT-traversing OSC sharing application OscGroups, by Ross Bencina, was chosen to be the central lynch-pin for the band's work. Using OSCgroups the band started to create compositions focused on their four laptops interacting in some fashion. These piece such as *Quartet for the End of Space*, a group controlled meta-instrument, were of increasing technical complexity and networking interplay. Due to OscGroup's unique NAT-traversing capability, this also marked the band's first experiments with performing in separate locations. To accomplish this the group revived the technique introduced to them at the RML gig with Chris Brown.

Each of the member's computers would locally render all of the sounds and visuals generated by each of the members control information output. This allowed for the members to perform in disparate locations without needing high-bandwidth internet connections. It also made it possible to have deeply interactive musical systems that influenced each other's musical output. While OscGroups did allow for NAT-traversing of fire-walls and routers, it did not always play well with academic networks. Furthermore, even with established connections, delivery of packets was not always guaranteed. This led to chaotic scenarios in which sounds or objects would exist on some of the member's computers, but not other's.

4.2. OSCthulhu 1.0

As the members left Oakland to pursue PhD research at separate institutions these issues of networking over large distances became of paramount importance. It was decided that a solution should be devised to meet the group's specific needs. It was for this reason that Curtis McKinney set out in April 2010 to create a custom made networking utility for the band, which would eventually come to be known as *OSCthulhu* (a play on OSC and H.P. Lovecraft's mythological demigod Cthulhu). This tool was heavily influenced by OscGroups, with its central task being the reliable delivery of OSC packets over the open internet.

A major structural decision was made to both increase the reliability of firewall/router penetration and the successful delivery of OSC packets: OSCthulhu would forgo OscGroups' NAT traversing mesh-network structure for a more traditional client-server model. Furthermore, OSCthulhu borrowed networking techniques from first-person shooters to increase the reliability of network traffic. By defining the entirety of the sound world as a *Sound-state*, the server could send synchronization messages at a regular interval to help combat lost packets, while maintaining use of musical gesture-friendly UDP. Version 1.0 was constructed as a java-based library, as at this point Java was the main programming language being used for compositional logic and visualization. This system proved to be much more reliable overall, and many successful concerts were given using it [8]. However, there were several issues. It severely restricted the implementation style for compositions, due to it being tied to a specific programming language. Furthermore the server itself became a point of failure, due to the possibility for secure-shell ssh time outs to cause the server end its process loop [1].

4.3. OSCthulhu 2.0

Version 2.0 of OSCthulhu was created to help solve the issues of version 1.0, as well as to add new capabilities. OSCthulhu 2.0 was programmed in C++ instead of Java to give it as little overhead as possible, and to make it more compatible overall [10]. Instead of being developed as a packaged library to be integrated directly into a composition's logic code, version 2.0 is designed to be a separate application that may be interacted with through an OSC based remote API. This provided a much higher degree of flexibility for implementation, as OSCthulhu 2.0 didn't restrict users to use Java.

The server architecture was also modified to be run as a daemon process on the open internet, so that OSCthulhu could be used at any point in time, without the need to rely upon an ssh session, which has the possibility of timing out, to run it. Extra features were added, including built-in chat functionality, an automatic reconnection scheme in case of disconnections, and the ability to broadcast OSCthulhu messages to multiple applications. The reliability of the system has allowed the group to create systems that require a high degree of interconnectivity and interactivity, with the simultaneous presentation of networked sounds and visuals, all the while performing from an extremely varied set of venues with differing technical setups. Due to the low bandwidth of the system, approximately 24 kilobytes/second on average, 3G phone networks may even be used in a pinch in lieu of a cabled internet connection.

5. CONCLUSIONS AND FUTURE WORK

Despite technical difficulties related to physical and temporal displacement, Glitch Lich continues to perform as a band. Much music and code has been written by the members and they continue to refine their technical skills. Other laptop bands looking to perform over long distances could potentially benefit from many of the solutions and procedures the band has adopted . Of key importance is a system to address both latency and information loss in the network.

Other important ideas include version control for code, precompiled binaries of software for all versions of operating systems the band members run, orchestrated simultaneous updates of key software, fail safe modes for complete loss of network connectivity, and beta testing performance software. However, technical issues cannot be the sole focus for a band. It is important to consider what makes network music unique and how to write for a disparate group.

All four members of the ensemble are doctoral students actively engaged in academic research. Much of this research focuses on issues in network music and often becomes a part of the Glitch Lich canon. Benjamin O'Brien is currently exploring interconnections between performers within complex hierarchies. His work bridges the gap between hyper-structuralism and free-improvisation. Chad McKinney has recently completed a new open source network music environment, Yig, that allows for the creation and manipulation of feedback matrices. He is currently developing new tools and techniques for network music environments including the usage of 3d worlds in performance. As the main contributor to OSCthulhu, Curtis McKinney is establishing the next generation of client and server architectures for musicians. He is also actively developing systems for cooperative sequencing and visualization. Cole Ingraham is currently investigating generative procedures using chaos in musical and visual contexts.

6. REFERENCES

[1] D. Barrett, R. Silverman, and R. Byrnes, *SSH, the secure shell: the definitive guide*, ser. Definitive Guide Series. O'Reilly, 2005.

[2] R. Bencina, 2010, available from: http://www.audiomulch.com/~rossb/code/oscgroups/ [Accessed 2 May 2010].

[3] J. Bischoff, "Unpublished e-mail to scot gresham-lancaster," 1998.

[4] D. Cope, *New directions in music*. Waveland Press, 2001.

[5] R. Fiebrink, G. Wang, and P. Cook, "Dont forget the laptop: Using native input capabilities for expressive musical control," in *Proceedings of the 2007 Conference on New interfaces For Musical Expression*, 2007, p. 3.

[6] S. Gresham-Lancaster, "The aesthetics and history of the hub: The effects of changing technology on network computer music," *Leonardo Music Journal*, 1998.

[7] C. McKinney, "Metagaming: Experiments With Art And Games," Master's thesis, Mills College, 2009.

[8] S. Reges, *Building Java Programs: A Back to Basics Approach*, 2nd ed. Addison Wesley, 2010.

[9] J. Rohrhuber, A. de Campo, R. Wieser, J.-K. van Kampen, E. Ho, and H. Hölzl, "Purloined Letters and Distributed Persons," in *Music in the Global Village Conference 2007*, 2007.

[10] B. Stroustrup, *The C++ programming language:*. Addison-Wesley, 2000.

[11] D. Trueman, P. Cook, S. Smallwood, and G. Wang, "Plork: The princeton laptop orchestra, year 1," in *ICMC '06: Proceedings of the 2006 International Computer Music Conference*, 2006.

[12] A. Ward, J. Rohrhuber, F. Olofsson, A. McLean, D. Griffiths, N. Collins, and A. Alexander, "Live Algorithm Programming and a Temporary Organisation for its Promotion," in *read_me — Software Art and Cultures*, O. Goriunova and A. Shulgin, Eds., 2004.

[13] M. Wright, 2002, open sound control 1.0 specification. Available from: http://opensoundcontrol.org/spec-1_0 [Accessed 2 May 2010].