ICE - towards distributed networked computermusic ensemble

Winfried Ritsch
Institute of Electronic Music and Acoustics Graz (IEM)
ritsch@iem.at

ABSTRACT

The deployment of distributed loudspeaker systems in the context of computermusic ensembles is explored in this paper, expanding the vision of a computermusic ensembles to a flexible dynamic musicians network.

With the IEM Computermusic Ensemble (ICE) the idea of playing in a virtual concert hall over network, rendered to one ore more real or simulated concert-halls has been explored: Each musician in the network is using virtual 3D-audio space to render and therefore position his/her musical contribution in this virtual concert hall, which can be rendered to the local loudspeaker system, like an Ambisonics spatial audio system, a binaural decoder, useful for remote musicians and monitoring and/or a projection of this in other spaces as audio installation. The musicians communicate with each other via musical audio data using the same network. In contradiction to most computermusic ensembles, ICE forms a distributed networked computermusic ensemble, able to play parallel and time synchronous in several spaces.

As an example the composition “All Under One Net” for ICE will be discussed and technical details, like the use of a audio message system, are explained.

1. INTRODUCTION

Traditional computermusic ensembles tries to embody computer musicians as universal instrumentalists, playing simulations of real instruments, augmented instruments, hyperinstruments [1], or sets of these instruments named before. The instrumentalists form locally an ensemble specialized in computer music and perform different music styles. Of course each ensemble has to focus on certain music styles and performances, but like at the beginning of electronic music in the 50s the “sound in unlimited space”[2] was visioned at the WDR Köln, here the vison of “musician with unlimited abilities” came up, restricted only to their restraints of implementations.

As an artistic research project, the IEM Computermusic Ensemble (ICE) will work on this vision and try to approach this in respect to our rich tradition in ensemble music and to extend the traditional view of computermusic ensembles towards a new kind:

mobility The instrumentalists can travel with small equipment and perform many different music styles as local musicians or distributed musicians over Internet.

distribution The play over network in an abstracted virtual concert hall (VCH), defined by interaction protocols, notation aspects and an abstract layer for spatial composition.

instantiations The VCH is rendered using a local loudspeaker system, like an 3D Ambisonics system higher order, in synchronous and at the same time in additional concert hall, binaural renderings or sound installations composed of these audio streams.

flexibility A computermusic musician can join and leave the ensemble dynamically, dependent on the nature of the composition like improvisations.

automation All parts in the chain of the ensemble, can be replaced by automation, to enable endless algorithmic compositions.

Therefore the vision of a computermusic ensemble is expanded to *“musician with unlimited abilities and boundaries”*.

As shown above, a special focus is directed to the usage of local loudspeaker systems, monitoring techniques and musician intercommunication on demand. Therefore one paradigm has been used: The music should stay as long as possible in the digital domain, so that any layer of performance can be distributed over network as audio or control data, either local on high speed networks or over the Internet with proper bandwidth.

2. COMPUTERMUSIC ENSEMBLES AND ICE

An computermusic ensemble as a group of musicians needs computermusic instruments, an concept about musician intercommunication for collective experience, how this ensemble is receipt over loudspeaker systems and/or network streams and a description of the capabilities of these musician with or without an real conductor.

2.1 Existing concepts

Besides singular ensembles using laptops, especially in the popular music and jazz, the need for laptop orchestras has been shown in publications [3] and discussed in Symposiums [4] since 2012.

Previous to this, programmatic playing with computer or calculators on stage in pop culture has been most famously shown by "Kraftwerk's Mensch-Maschine concerts[5] since 1978 and others like with the album ISDN by " The Future Sound of London"[6]. Previous to this time also experimental music bands started using computers on the stage in concerts.

Pre-Internet networked concerts as experimental art events hast been done at least since the 90s with slowscan-tv, fax and MIDI over modems [1], where as intercontinental concerts, musicians tries to combine different concert spaces to a shared space in the net. With the availability of high speed Internet the possibility of merging concert spaces via multichannel video and audio has be shown within the CO-MEDIA project within Europe[8][9] to work.

Besides the "electronic musician" since the 50s, "laptop-musicians" sicne 80s became common in most areas of music performance practice. In the following some aspects of computermusic ensembles are discussed in respect to new concepts for ICE:

SLORK, Stanford Laptop Orchestra [10]: musicians use special loudspeaker for localization and distribution of sound located near the musician, to be spatial matched by the audience.

L2ORK, Linux Laptop Orchestra [11]: usage of a customized flavor of Pure Data [12] on uniform computer sets. The use of a network play an important role for control of the performance by a conductor. They also restrict the interaction interfaces with a unique sensor setup and act like a choir.

PD˜Graz [13]: (for example playing "blind date"[14]]) is a group of computer-musicians playing together on one, for multiuser usage modified, version of PureData as a conjointly playground. The audiovisual performances are played by a varying amount of players, introducing "live coding"[15] as improvisation method. With this they use a common virtual computermusic space with their laptop computer as interfaces.

The resulting research questions for a new concept of an computermusic-ensemble are investigated below:

computermusic instrument design How the sound of the designed instruments are assigned to instrumentalists by the listeners ?

interaction interfaces How is the embodiment of these instruments on stage ?

The selection and mapping of sensor interfaces for expressive playing of the instruments is important for the match of their performance with the sound of the instrument.

interaction between musicians How they receipt and distinguish each other ?

Additionally the synchronicity of their data streams, including the automation of their interaction is important to form groups and sub-groups.

Spatiality How should the musician use the speaker system to correlate the spatial perception of the instrument output with each musician playing ?

[1] Experienced by the author as musician within Art's birthday concert: Innsbruck Austria - Vancoucer Canada on 17.1.1993 [7]

Figure 1. L2ORK concert (at citadel) versus PD-Graz performing blind date

conduction and composition What are the roles of the conductor and how to write scores for him ?

The conductor can be implemented by means of software, as a real conductor or a combination of them as a computer aided conductor.

dissemination How should the capabilities of these ensembles be disseminated to composers, musicians and audience ?

Writing compositions can include programming some aspects and giving the musician the ability to train and rehearse for a piece.

2.2 The vision of ICE

The IEM Computermusic Ensemble ICE, has the aim to be an ensemble of computer musicians with an optional real or virtual conductor, playing with their individual computer setup in an network to a virtual concert hall [2] (VCH), which is rendered to one or more real electro-acoustic concert hall or rehearsal space.

Instead of local amplification, microphoning and/or direct feed of an loudspeaker system, the musicians play together within VCH as a 3D-audiospace, distributed over the network, The VCH can be seen as an abstract virtual playing space. For spatialization either encoded material or streams of audio with spatial data is used to represent

[2] "Virtual Concert Hall" in the sense of simulated concert room in 3D, which has been realized in a frist attempt within the e-learning project Internet Archive of Electronic Music VCH-IAEM at IEM 2002 by the author using the Ambisonics domain as an abstracted representation to be rendered over binaural decoders or a real multi-speaker environment[16]

this domain, There is an rendering engine for the 3D spatialization data for every instantiation of this virtual space, like a real concert hall with an multichannel audio system or binaural decoder for headphones. This makes the instrument design independent of the sound-systems. The the sound-system should provide certain features, like 3D-spatiality (at least 2D). The quality of the concert depends therefore strongly on the quality of the 3D audio system.

An instantiation could be an Ambisonics speaker systems or any other 2D- or 3D- rendering engine available as long it can map ambisonics to it [3] . Out of this, each musician use the this speaker systems as projection space for his contribution, not a local speaker nearby. His instruments are independent from the actual setup of the loudspeaker system.

Due to the fact, that the VCH can be streamed over network and each musician send streams his contribution into this virtual space, there is no difference of access for the remote or local musician. Using binaural renderings of the concert hall at home or a studio, or even playing in an other concert space at the same time over network is possible and physically distributed musicians can form one computermusic ensemble. Additionally they can interact with each other over network individually.

Also the ICE computer-musician is not depended on unified hardware, software or operating systems, but on standardized transmission protocols for audio and control data over a high speed network. All the audio and control data transmission are done over predefined protocols and streaming formats. The spatialization engine could be controlled by the real or automated conductor has has to be defined within the composition.

Figure 2. first concert of IEM computermusic ensemble ICE playing over a HUB

Looking at ICE to establish an ensemble with local or remote computermusic musicians, playing in front of the audience like an chamber music ensemble, computer-musicians play either simulations of real instruments or experimental virtual instruments. These have to be designed for the compositions targeting ICE.

[3] VBAP, wavefield can simulate an Ambisonics speaker set, the mapping in the other directions is much more complex

A main goal is to enable the creation of new compositions with a plausible reception of ICE as an ensemble, reflecting also the compositional and sound style of our Institute and associated artists, establishing a kind of a western style computermusic on one side, as also perform each kind of standard ensemble music.

Do accomplish this practically, a number of issues has to be solved in this respect and the development of tools has to be done for the needs of this purely computermusic ensemble. Defining the standard networking protocols invented the ICE-Bus.

The overall vision could be summarized with the following mission statement:

> The IEM Computermusic Ensemble (ICE) is a group of local or remote musicians, specialized playing with computermusic instruments, such as interactive, augmented hyper-instruments or simulations of real instruments. The main target art is contemporary music adapted or written for computermusic ensembles, but also interpretation of historical ensemble music with virtual instruments can be performed. As a speciality, they play with a 3D-spatial audio systems, especially Ambisonics, to be rendered in small or large concert halls or virtual environments at the same time and perform distributed network concerts.

3. THE ICE-BUS

The ICE-bus is a umbrella development as a kind of specification of protocols with some example implementations as start, to be expanded over the time enhancing the ICE. As art driven development it was deduced from actual demands and compositions and verified within implementation in art projects.

One thing is the acceptability of ICE as an ensemble by the audience, another the practically implications on the need to fit the current practice of performing music in different locations. Fast setup, stable operation and lightweight equipment are important premises. Additional the ability to join in the performance as musician over the network dynamically from all over the world, is an important issue for some compositions. When specifying the audio-network for playing together within the ensemble, a focus was set on the collaborating efforts to be done to gain the unity of the individuals.

Figure 3 shows a simplified draft for data network structure of the ensemble. One unique feature is, that the fast Ethernet connection, preferred wired, is the only hardware connection between the musician and between musician and the conductor besides the power supply.

The conductor, seen as musician $M0$, is responsible for rendering the abstract spatial audio signal to the speaker system and is a real, partly virtual musician as a software which conducts and renders the performance. He takes control of the local audio rendering engine, at least in the sense of directing the local and/or remote music contributions. He could be backed by a technician.

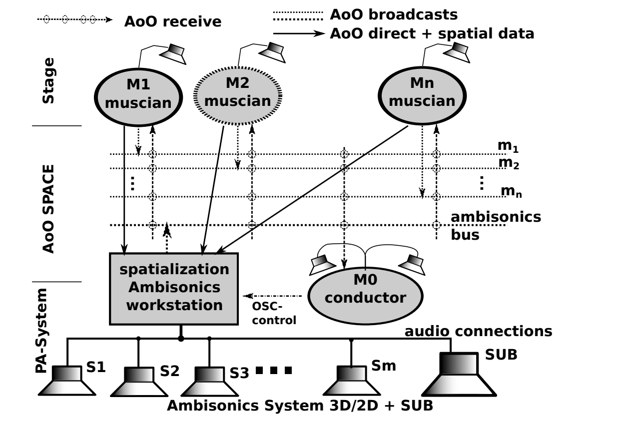

Figure 3. ICE using network connections for intercommunication and playing on to shared sound system

As described before, with this 3-D audio bus, every musician can spatialize him- or herself to the sound system. There is no need for individual loudspeaker of musicians to be heard by others, but each musician needs some possibility for monitoring his and maybe the neighborhood audio signals, the output of the VCH and click tracks. This monitoring signal is extracted from the ICE-Bus by performing an individually monitoring mix for each musician by each musician. Mixing and leveling individual musicians for the performance is done by the conductor or under his supervision, extracted from the ICE-bus.

Global messages like beats, start, stop, triggers oder sensory data is exchangeable. Anyhow every timeless or timed data used for networking should be described within the score of the piece.

This enables the musician to inter-operate more easily, the conductor to broadcast data to musicians and the composer to use these protocols as a kind of notation, such that the piece can be archived for future interpretations.

Looking for a modern, commonly used transmission format for messaging systems within the computermusic domain, we chose “Open Sound Control” (OSC) [17]. With its flexible address pattern in URL-style and its implementation of high resolution time tags, OSC provides everything needed as a communication format[18].

3.1 Audio over OSC (AoO)

Experimenting with static network-streams, e.g. using jacktrip [19], showed good results, but they need high bandwidth and are inflexible within complexer networks, especially distributing to multiple listeners on the ICE-bus since mostly it cannot be send as broadcasts. Also in case of server or client breakdown, the whole network tends to break down. Since also a lot of messages synchronized with audio steams are needed, it seems to be more simple and straight forward to handle audio contributions of musicians the same as messages. As a conclusion the decision to use OSC (Open Sound Control)[18] also for transmitting audio was done. As an solution to that problem AoO was developed, which has been published at the Linux Audio conference 2014 and is described there more precisely[20, 21].

With *AoO* the participant just needs to connect to the network, wireless or wired, choosing the drains to play to and send phrases of audio only when needed.

For ICE Ambisonics was chosen as the abstraction of the virtual audio environment. Within Ambisonics domain, each musician can always use the same playing parameters for spatializing the musical contribution. With the imagination of the musician playing in the VCH, as ”a virtual 3D environment“, they send either their audio signals together with 3D-spatial data or a pre-rendered Ambisonics signal to the dedicated drains. This rendering system act like a distributed mixing system, which produce an Ambisonics mix.

Additional there is the possibility of audio communication between the musicians. Each musicians can broadcast the produced audio signal and hear into the signal produced by the other. Also there is the possibility that musician can send, on special offered drains by the receivers, dedicated audio intervention to the others.

The musicians have to do their own monitor mix, depending on the piece and space where the play, listening only to the neighborhood musicians or the whole audio-scene.

The advantage using a message audio system is, if each musicians sends only sound data if needed, like audio bursts or rendered notes or sending their audio-data to other musicians bandwidth requirements can be dramatically reduced. There should be no border on the imagination of complexity of the ICE network to be used, as long it can be grasped by the participants.

For ICE a first library was written at the IEM using AoO[21], also addressing some problems with asynchronous data transmission.

3.2 Message Bus

For the ICE-Bus a complete set of control messages has to be specified, using a well defined namespace for the ICE-Protocol. This has been tested in first concerts and was specific to the first pieces of ICE implementation, but is and will be expanded if needed for other compositions, as long as they guarantee upward compatibility and proper ”running code“ is provided for the ICE-Bus.

As in most ensembles, everybody has to trust each other. Therefore no special need for security are enforced, to simplify the setup. One exception is addressing the local sound-system, which can harm listeners, so this drains should be protected to be used exclusively by the main rendering machine.

One important aspect is, that most of all messages are sent as broadcasts, so every musician is able to filter the relevant messages for him or her and the conductor can send musician messages as overrides. For example, controlling the spatialization is primarily a task of the musician himself, but can also be overtaken by the conductor for specific pieces.

Furthermore, a set of different sensors interfaces should be usable for each musician, depending on the instrument or instrument class used. For example, a drummer has a kind of tracked sticks or a string-player accelerators and so on, but the controlling messages for the sound generation should have the same parameters, using an abstracted parameter set. Mapping controller data to these control parameter is in the responsibility of the musician. To embody the performance the computer-musician should avoid traditional computer interfaces, like sitting with keyboard and mouse in front of a monitor, as much as possible. With fulfilling the general syntax definitions of ICE, an abstract layer within the OSC-namespace is shared between the musicians.

For example: As special interfaces, transcription engines could be used. Transcription means, for example using voice to control a flute, or using other sound input to play strings or doing gestures mapping for modulations. This has been shown within the piece "Maschinenhalle #1", where metal plates are interfaces of dancers to player pianos [22]. This gives more possibilities to interpret special pieces, with different interfaces. Therefore the sensory layer is individual to each musician and will be implemented and adjusted by the musician who uses them.

For OSC messages REST (Representational State Transfer) style is used, like described in "Best Practices for Open Sound Control"[18]. With its stateless representation each message is a singleton containing all information needed.

Historical seen, orchestras function in a strong hierarchy. Here we use this kind of hierarchy for structuring the control data. The hierarchy is from top level down to instruments functions:

```
/ice/<id>/m/<nr>/{instr/<instr>/...|<sm>...}
    [args, ...]
id ... ensemble name or number
nr ... number of musician from 1..N
instr ... instrument name or number
sm ... system message
```

Since more than one ensemble can play on the same net, we have to distinguish them with the id. Musician nr is 0 for the conductor and $1..N$ for musicians, assigned in the score. All messages should be send via OSC-bundles to enable time-tagging.

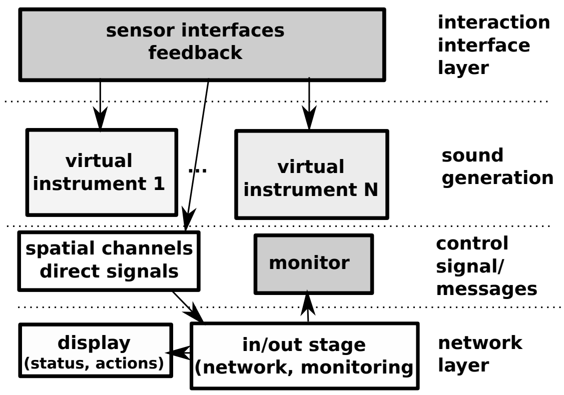

Figure 4. software layers of the musician instrument

3.3 ICE software library

The ICE-Bus software library is a framework for performing pieces. For ICE composition is also coding, composers have to use parts in the layer structure to easily integrate the pieces in a repertoire. Musician has to take care of the specific musician depend modules: Individually different for each musician is the sensor layer which consists of the interaction interface for the sensors the musician is trained for. It should generate a defined set of control parameter to drive the generative part of the instrument, which is needed for the specific composition.

Another part to be altered by the musician is the monitoring, because each musicians needs other strategues to monitor during playing their specific interaction interface.

The instrumentation is done with virtual instruments for each piece, which are driven by interaction interface as abstract functionality, e.g. $/in/sensor/wii/X$ to $modulation$, which has to be implemented by the musician individually.

Rendering in the abstract 3D concert space is done with a the audio processing library used by all musician to gain compatibility. Some parameters has to be controlled by sensors, depending on the piece. So there is a clear distinction for which part the musician, the composer and the system integrator is responsible for.

The interfaces in this software library will be published as documented code, so each part can collaborate within the composition process of a piece. For example a composition addresses musician to play different frequency bands, the functionality would be implemented in the instrument section by the composer. If a musician plays another instrument, the abstracted parameter interface between interaction interface and instruments should match the needed parameters.

3.4 synchronization and time

As a general time, tempo, bars and divisions are used to gain the conductor to control the speed within the piece. If fixed time is needed in some pieces, a synchronized realtime clock can be used in the composition. It depends on the piece if audio has to be phase synchronous or not. Since jitter for messages is mostly more worse than a constant latency, it is recommended to use a central timesource. Available for most system, the "Precision Time Protocol" (PTP)[23] allows a lightweight implementation in local (and remote) networks and can guarantee a quasi sample-accurate synchronization. For distant network the "Network Time Protocol" (NTP)[24] can be used, since it is the standard for the Internet and distributed over routers and gateways. AoO provides this kind of synchronization as module.[AOO:14]

4. ALL UNDER ONE NET

As an example piece, "all under one net" (auon) shows some how ICE is implemented and some important new aspects of distributed computermusic ensembles.

The musicians play with 1st-order Ambisonics-recordings as sound material. This contributions are also rendered in realtime in the 3D concert space. They are also sidelined into a central algorithmic compositional machine, which output will be mixed to contributions.

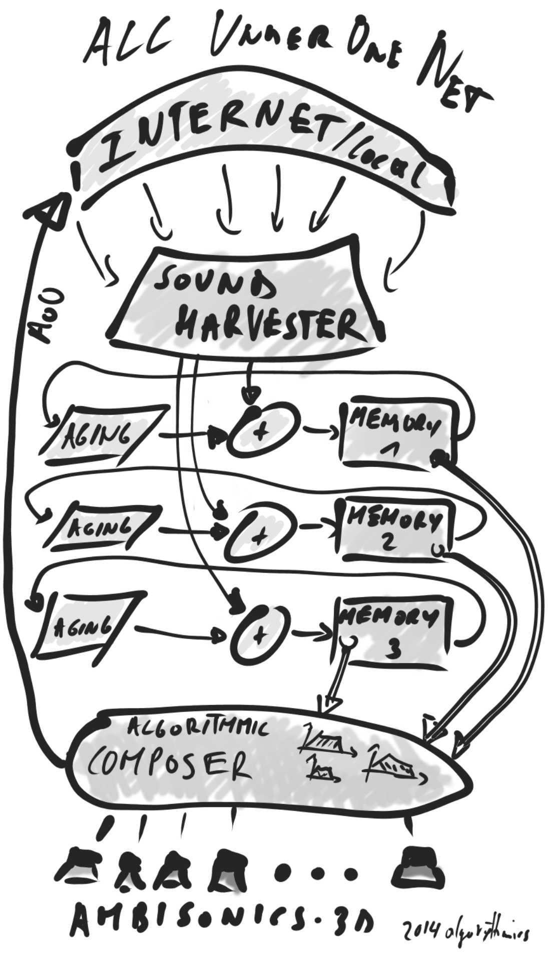

Figure 5. graphic score All Under One Net for ICE

4.1 algorithmic composer

The algorithmic composer, simulates short time, medium time and long time memory, aging material with filters, cutters and pitch-shifter in Ambisonics domain. Ambisonics material is categorized for feeding the different memory stages. As an outcome the collected material is altered (aging) continuously over time in the three memories spaces, implemented as loops. The algorithmic composition generator machine, using self-replicating data structure from an initial series, extract glimpses from the different memories and spatialize them in the abstracted 3D concert space.

4.2 rendering to the sound system

Musician play Ambisonics soundscape material, recorded from typical places of their origin, direct in the Ambisonics space. With the new technologies, like the AmbiX toolkit [KRON:14], which can stretch, expand and enhance 3D-material on different directions, the contributions of different musician can be spatialized surrounding the audience. Generating focuses as main direction for each musician, it is possible also to use full 3D-Ambisonics material the same as instrumental sounds with coordinates. Mixed with the output of the algorithmic composer, which can be rendered more in distance and from the back. Musician can also monitor the whole sound scene using binaural Ambisonics rendering plugins with headphones, if they are positioned outside the main listening space, like on a stage or remote musician.

4.3 networked remote musicians

All Network places are equal, except of one where the conductor is placed as musician M_0. The conductor arranges the spatial perception of the musicians and also provides the timing. The local Ambisonics signal is send to the remote locations and mixed with the remote ones. To reduce bandwidth, musicians contributions are not send to remote, since monitoring should be done over the VCH represented by the Ambisonics mix.

4.4 performances

A first version was composed for ICE in 2011 and performed at Porgy an Bess in Vienna [4] and at the IEM. The main focus was to explore if ICE is perceived as an ensemble and if a "not computermusic specialized audience" can follow the musician and assign the receipt audio material to the musicians. Besides being recognized as unusual musicians, most people in the audience confirmed both.

Figure 6. remote screens in vilnius of CUBE in Graz

In a next Version 2012, the network concept was added, playing with musicians in Vilnius, rendering Ambisonics 3.order on both concert halls. With a fast Internet connection, streaming 25 channels audio and 2 video-streams in each direction with latency about 60 ms on audio including local rendering machines, most pieces worked well, especial aoun. Rhythmical pieces lack on synchronization by conduction over video, where latency of the video with at least 3 frames is much bigger. Conducting over OSC worked much more reliable.

As a new version, the spatialization of of Ambisonics material with techniques of enforcing directions with spatial

[4] A well known JAZZ club in Vienna

compressions[KRON:14] for better separation of material. Within this Ambisoinics soundscapes different order can be spatialized in a concert hall enforcing different directions.

5. CONCLUSIONS

After of 3 years of experience, we recognized the potential of this concept and therefore decided to publish it as open source toolkit and tray to trigger more compositions for this setting. Also the tools, especially AoO, can be used for other applications. As a conclusion this project will be enforced in future and hopefully more remote places will be added.

Acknowledgments

Thanks to Wolfgang Jäger for a first implementation of AoO as a sound-engineering project, Matthias Kronlachner especially the remote concert deployment. Special thanks for helping to bring ICE onto stage by the members of the ICE ensemble, especially to my students putting extended working energies in this project, patiently work with untested software and extend the sensor layer a lot. Also thanks to all providing us concert spaces.

6. REFERENCES

[1] T. Machover and J. Chung, "Hyperinstruments: Musically intelligent and interactive performance and creativity systems,," in *ICMC Proceedings 1989*.

[2] H. Eimert and R. Beyer, "Klang im unbegrenzten raum," CD, 1990, track 1, 1952.

[3] D. Trueman, "Why a laptop orchestra?" *Organised Sound*, vol. 12, pp. 171–179, 8 2007. [Online]. Available: http://journals.cambridge.org/article_S135577180700180X

[4] T. Soule, Ed., *GECCO '12: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2012, 910122.

[5] P. Bussy, *Neonlicht - Die Kraftwerk Story*. Bosworth Edition, Berlin 2005, 2005, p. 110ff.

[6] Wikipedia, "Isdn (album) — wikipedia, the free encyclopedia," 2014, [Online; accessed 15-July-2014]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=ISDN_(album)&oldid=612355144

[7] H. Bull, "Midi-jam,zero project in innsbruck," http://www.artsbirthday.net/1993/, 1993, online accessed 10.4.2014.

[8] F. Ircam, "CO-ME-DI-A, EACEA Culture Project on Network Performance in Music," http://www.comedia.eu.org/, 2009, [Online; accessed 1-Feb-2014].

[9] "Preserving sound source radiation-characteristics in network-based musical performances," 2011, dAGA 37. Jahrestagung für Akustik, 21.-24.03. Düsseldorf, Germany 2011.

[10] S. University, "Excerp from Bio," http://slork.stanford.edu/, 2011, [Online; accessed 7-May-2011].

[11] D. L. L. Orchestra, "An Excerpt from Ico's L2Ork Ramblings on the vision of L2ORK," http://l2ork.music.vt.edu/main/?page_id=5, 2011, [Online; accessed 7-May-2011].

[12] IEM, "Realtime Computermusic Language from Miller Puckette," http://puredata.info/, 2011, [Online; accessed 7-May-2011].

[13] P. Graz, "Pd˜Graz," http://pd-graz.mur.at/, 2005, [Online; accessed 7-May-2011].

[14] ——, "Blinde Date," http://pd-graz.mur.at/blind-date, 2011, [Online; accessed 7-May-2011].

[15] I. m zmölnig, "Live coding: an overview." in *Proceedings of the ICMC, Kopenhagen*, 2007.

[16] "Internet archive for electronic music - internet audio rendering system iaem - iars." Springer Verlag, Computer Science Series (LNCS 2771), computer Music Modeling and Retrieval 2004 Conference.

[17] M. Wright, "The open sound control 1.0 specification." http://opensoundcontrol.org/spec-1_0, 2002, [Online; accessed 1-Feb-2014].

[18] A. Schmeder, A. Freed, and D. Wessel, "Best Practices for Open Sound Control," in *Linux Audio Conference*, Utrecht, NL, 01/05/2010 2010.

[19] J.-P. Caceres and C. Chafe, "jacktrip," 2008, [Online; accessed 12-Dez-2011]. [Online]. Available: https://ccrma.stanford.edu/groups/soundwire/software/jacktrip/

[20] W. Jaeger and W. Ritsch, "AOO," http://iem.kug.ac.at/en/projects/workspace/2009/audio-over-internet-using-osc.html, 2009, [Online; accessed 12-Dez-2011].

[21] W. Ritsch, "Towards a message based audio system," in *Proceedings of the LAC 2014*, 2014.

[22] ——, "sound plates as piano interface," in *Renew/IMAC 2011 Proceedings*, M. Sondergaard, Ed. Aalborg: Aalborg University Press, 2012, p. 21ff. [Online]. Available: http://vbn.aau.dk/files/73765571/IMAC_2011_proceedings.pdf

[23] T. C. on Sensor Technology, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, ieee std. ed. New York: The Institute of Electrical and Electronics Engineers, Inc. (Hrsg.), November 1588–2002.

[24] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905 (Proposed Standard), Internet Engineering Task Force, Jun. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5905.txt