UbiMus2023 Proceedings

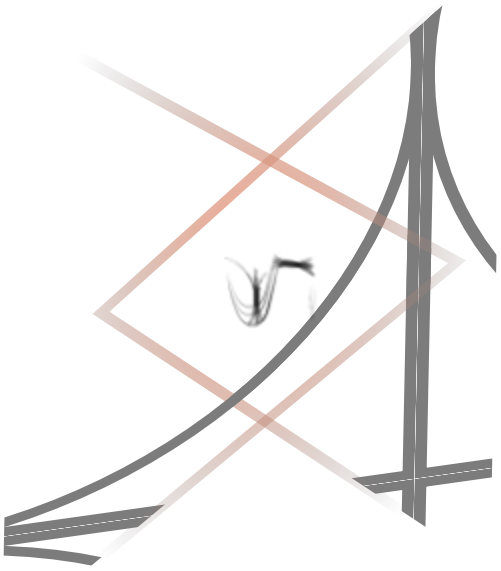
Edited by Azeema Yaseen, Brian Bridges, Marcello Messina, Damián Keller

Ubiquitous Music (UbiMus) 2nd International Symposium

November 2–4, 2023 (In-person and Online)
Ulster University, Derry~Londonderry Campus, Northern Ireland
https://www.ulster.ac.uk/conference/ubimus

Ubimus2023 Youtube

ISBN 978-65-00-85069-7



UbiMus2023 Symposium

November 2–4, 2023 (In- person and Online)
Ulster University, Derry~Londonderry Campus
&
Keynote Talk and Performance at North West Regional College

UbiMus2023 Symposium recordings are available here:

youtube.com/@Ubimus2023UlsterUni

Supported by:

Connected (HE-FE fund)

Organising Committee (Ulster University, NWRC, and Ubimus partners)

Ulster University and North West Regional College (NWRC)

Dr. Brian Bridges (Ulster University): Chair, Co-chair of Papers
Dr. Rob Casey (Ulster University): Co-chair of Music, Artworks and Workshops
Dr. Karl McCreadie (Ulster University): Co-chair of Papers
Dr. Adam Melvin (Ulster University): Co-chair of Music, Artworks and Workshops
Mr. Tony Talbot (NWRC): Co-chair Co-chair of Music, Artworks and Workshops
Mr. Aaron McGlinchey and Mr. Patrick Moore (Ulster University): Technical support
Mrs. Caroline Harkin (Ulster University): Administrative support

Ubimus Network and Proceedings Editors

Dr. Damián Keller (Federal University of Acre, Brazil): g-Ubimus Coordinator
Ms. Azeema Yasseen (Maynooth University, Ireland): Editor of Proceedings
Dr. Marcello Messina (Southern Federal University, Russian): Co-editor of Proceedings
Dr. Teresa Connors (Concordia University, Montreal): Reviews Coordinator
Prof. Victor Lazzarini (Maynooth University, Ireland): g-Ubimus network representative

The local organisers would like to thank:
Ms. Karen Reid: Ulster University Research Impact team
The Connected **HE-FE** collaboration fund

Preface

It has been an honour and a pleasure to chair UbiMus2023, the thirteenth Ubimus event (previously styled the International Workshop on Ubiquitous Music), at Ulster University and North West Regional College, which we hosted in the city of Derry~Londonderry, Northern Ireland's second city (and the fourth-largest city on the island of Ireland).

Ubiquitous Music is an interdisciplinary research area that combines methodologies from music, computer science, education, creativity studies, human sciences and engineering. The international nature of the Ubimus network has led to our receiving papers from researchers based across South America, Europe, Asia, and Australia, with researchers themselves reflecting even more regional and national diversity.

It is therefore befitting of this diversity that we held this event in the city of Derry~Londonderry, situated at the border between the UK and the EU. The socially-engaged nature of Ubimus also chimes with Derry's history of engaging with issues of social justice, both nationally and globally, having been at the centre of the Northern Ireland Civil Rights movement in the 1960s and 1970s.

More broadly, a much-needed awareness of diversity has come to the fore within arts-technology fields, and Ubimus is no exception. To this end, we were delighted to have our second invited lecture, on Aural Diversity, by Prof. Andrew Hugill (University of Leicester, UK), who has recently led a research project in this area. The support of the Northern Ireland Department for the Economy made it possible to connect Ubimus with Africa for the first time in such an event, bringing our keynote Wanjiru Ngure, alias [M] (formerly Monrhea) to present on her approach to live music coding, informed by her work (both as a performer and workshop leader) within the Nairobi music scene.

This hybrid (online and in-person) Ubimus event allowed us to develop new music and workshop strands, and I'm particularly grateful to the artists, musicians and composers who have given so generally of their time and creative work. As I was compiling the recorded symposium videos, I was once again struck by how vibrant this research and practitioner community is. In spite of the challenges of hybrid conferences, the experience of connecting so many continents and countries, through both virtual and physical presence, was a uniquely satisfying one, and this is before we consider the workshops, artworks, and the multichannel audio concert, which, unfortunately, did not fully lend themselves to remote presentation (though if you were participating remotely, you can find out more about them in the programme).

It was a delight to be present with you for this event, whether in-person or online. I'd like to pay a particular tribute to those who had the farthest to come to be with us in person: our keynote [M] [Monrhea] – Wanjiru Ngure, our workshop co-leader Nyokabi Kariũki, and Gerald Estadieu, who, between them, connected four continents. In the virtual space, I'd like to thank Prof. Andrew Hugill, who delivered such a thought-provoking Invited Lecture, which informed so many follow-on discussions. I'd also like to thank Damián for his support and leadership within the network, and Victor and Stephen for their support in chairing sessions. This event wouldn't have been possible without my Ulster and North West Regional College colleagues (you can find the full credits list elsewhere in these Proceedings), and we are also very grateful to the Connected fund (for HE-FE collaboration).

The support of colleagues around the review and proceedings editing was also crucial; sincere thanks to Teresa for helping to coordinate the review process, and to Azeema and Marcello for their lead on editing the proceedings. Thanks to everyone who participated, in whatever manner or capacity, in support of this event.

I hope that this year's symposium has provided opportunities to interact and draw inspiration from our shared work. On behalf of the local organising committee, including my Ulster University colleagues from Music (Drs Rob Casey and Adam Melvin) and Computing (Dr Karl McCreadie), and North West Regional College (Mr Tony Talbot and Ms. Grainne McNamara), I would like to close by saying that we were delighted to meet you all here in Derry, and look forward to connecting at future events within this space.

Dr Brian Bridges

Ubiquitous Music as a Movement?
(UbiMus 2023 Editorial)

Damián Keller,[1,2] Marcello Messina,[1,2,3] Brian Bridges,[4] Azeema Yaseen[5]

[1] NAP, Federal University of Acre, Brazil
[2] NAP, Federal University of Paraнba, Brazil
[3] Southern Federal University, Russian Federation
[4] Ulster University, Northern Ireland
[5] Maynooth University, Ireland
Ubiquitous Music Group
November 2023

Ubiquitous music practice has expanded from a core set of research groups, mostly based in Brazil, to a highly diverse and globally distributed intercontinental community, with a recent development being the addition of practitioners and musicians from Asia and Africa. For this year's symposium, we were particularly happy to be able to invite our keynote Wanjiru Ngure, alias [M] [Monrhea], to present a public lecture and workshop on her approach to live music coding, informed by her work as a performer and producer, and also as a workshop leader, within the Nairobi music scene.

The geographical diversity noted above is mirrored in the range of aesthetic and conceptual approaches featured within our community. It is precisely due to this richness in cultural and epistemic approaches that we were particularly happy to welcome this 2023 edition of our Symposium to Derry (also known as Londonderry), a border city marked by unique geographies and histories of resistance, activism and coexistence; a city that encourages us to say that ubimus is also about self-determination, pluralism, and — something we cannot refrain from wishing for in the conflict-ridden days of late 2023 — meaningful communication and coexistence, i.e. global peace!

Throughout the *first wave of ubimus* initiatives (2007-2014), the theoretical and methodological undertakings could be described as "patchy" or otherwise messy, and exploratory. One typical criticism of ubimus proposals was the lack of an explicit definition of the field. This topic fueled multiple exchanges among ubimus researchers, sometimes with additional contributions from those outside the ubimus community who engaged with the field's concepts in their own debates; e.g. (Cullimore and Gerhard 2015) on virtuosity in human-computer interactions for composition. A gradual, but increasingly solid, consensus started to form around a notion of ubimus as an ecology or ecosystem of stakeholders and resources supporting creative endeavours (Keller and Lazzarini 2017).

As we reach a decade of what might be described as *second-wave ubimus*, viewing ubimus as a movement rather than as a strictly bounded discipline or set of related research projects may provide us with a useful conceptual grounding, highlighting features that set it apart from mainstream approaches to music and computing. One aspect is the type of materials targeted within ubimus practices and discourses. Mainstream tendencies tend to enforce a restrictive definition of music that separates "sonic" from "musical" materials. This split implies a tendency to focus on symbolic representations of musical materials, leaving the "dirty little secrets" of sound shaping or timbral modelling to specialised

techniques. As a result, we sometimes see a knowledge gap between musical interaction (mostly understood as the construction and usage of instruments or of emulations of acoustic-instrumental interfaces) and sound art (a set of practices focused on audition as opposed to other modalities of perception). Ubimus practice does not encourage these separations; indeed, at a conceptual level, emerging ubimus frameworks question the utility of these boundaries; see (Messina et al. 2023; Mesz et al. 2023) in this collection.

Another feature of ubimus that seems to abide by the notion of movement is the coexistence and complementarity of multiple (conceptual and procedural) frameworks. Alternatively, this may also be a characteristic of a second-wave ubimus definition-building phase that may eventually result in emergent overarching approaches to design, deployment and validation of technological support. Three tendencies have been identified as threads that combine diversified research practices: *dialogics* (Lima et al. 2012), *ecologically grounded creative practice* (eco) (Keller and Lazzarini 2017), and *computational thinking* (Otero et al. 2020). Some aspects of the dialogical approach are compatible and complement the eco approaches. Whether these two threads remain separate depends on the outcomes of the field deployments and on the identification of specific phenomena that may require separate or composite treatment. Since Otero and coauthors' formulation of computational thinking as a potential conceptual tool for ubimus, several coding techniques have emerged that show very promising results; e.g. (Kramann 2023; Lazzarini and Walsh 2023; Zheng et al. 2023) in this volume.

Another characteristic of ubimus research is the exploration of alternative avenues for creative practice. This is a subtle yet provocative notion, because it also implies the design of new strategies to assess creative procedures and products. It could be argued that innovation is just in the eye of the beholder (or in the ear of the listener) and this person-centred approach is one of the reasons why ubimus methods have increasingly favoured field deployments involving diverse contexts.[1] This tendency also points to the need to validate procedures not only *within* communities but also *across* communities. Whether this is possible, and how this validation could be enabled, are still open questions.

Thus, the search for an expansion of creative horizons, the coexistence of complementary ubimus frameworks and the resistance to establish explicit boundaries among the potential materials to be employed in music-making all point to a convergence toward aesthetic pliability as a common target of second-wave ubimus perspectives. How to achieve this goal, taking into account the local cultural factors and the material constraints demands extensive field deployments. The variety of artistic outcomes showcased in the 13th Ubiquitous Music Symposium, in papers as well as in the artistic programme, is an indicator that our community has come a long way toward this goal. Let us consider a few of the areas addressed in the proceedings.

(1) *Ecologically grounded creative practice and tangible interaction* are avenues of investigation which are actively being explored within our community. Despite the expansive potential of these perspectives, some projects may still carry the remnants of a narrow interpretation of the E4-cognition framework (ecological, embedded, embodied, enactive cognition). This is implicit in the notion of body movement and instrument usage as the *only* valid strategies for musical interaction. Acoustic instruments and their emulations do have a place in ubimus. However, ubimus

[1] There is a conceptual gap where HCI methods attempt to meet creativity. Our community has tried to articulate this issue around the idea of aesthetic interaction in ubimus. But we still lack a discussion of the *computationally creative* elements. These are not included in "human" or "person" centric interaction.

frameworks for musical interaction do not depend solely on acoustic-instrumental models, and the conceptual and creative restrictions implied by a rigid or normative adoption of gesture-based frameworks, or an implicit adherence to acoustic-instrumental models, may limit developments within the field if we are not conscious of the conceptual baggage which accompanies them. In the context of this year's proceedings, Duarte's (2023) work in blending sonification, sculpture, and metaphor in his piece *Augury* provides an alternative to prescriptive models of embodiment via his use of non-Eurocentric and varied historical concepts of divination, resulting in an extended and vividly multi-modal experience (including synchronised lighting and visual diffusion effects) fostering audience engagement with the work.

(2) The strategies for *prototyping*, involving the application of *computational thinking*, are rapidly expanding. *Arithmetic Operator Grammar* is a generative technique that has been devised as a simplified approach to handle dynamics, pitch and duration parameters (Kramann 2023). Its most recent iteration features an environment that provides support for short scripts that yield carefully constrained musical outcomes which may be characterised as "AOG sounding". Complementarily, *lite coding* is proposed as a simplified form of programming that may be accessible to untrained subjects (Zheng et al. 2023). Both approaches highlight the usage of a limited set of symbols to generate interesting musical outcomes that do not demand extensive technical training.[2]

(3) This perspective on simplified prototyping may also be applied to the realm of *digital signal processing*. Very interesting possibilities are unveiled in three independent projects that point to an evolving ecology of interrelated tools implemented by the ubimus groups based in Ireland. Jagwani (2023) discusses the prototyping support afforded by the recent implementation of audio processing and synthesis languages. Chaiphas (2023) employs Cabbage and Csound to design a binaural audio processing tool. Lazzarini and Walsh (2023) present two brand-new computational environments that support highly efficient audio processing (Aurora) and visually based prototyping (Lattice). This body of initiatives attests to the potential of interaction and interoperability between ubimus projects.

(4) *Gastrosonics* is an emergent trend within ubimus which finds an alternative way to explore musical experience in *multimodal and cross-modal* contexts (Keller, Alcвntara-Silva and Mesz 2022). Using food and drinks to trigger aesthetic choices and alternatively employing sound to shape gastronomic experiences are two strategies that may be fruitful for future ubimus research. Mesz et al. (2023) highlight potential avenues for further investigation in artistic and research-driven developments within ubimus.

(5) As previously stated, a key goal of ubiquitous music research is an *expansion of our understanding of creativity*. This issue has been approached both as a theoretical challenge and as a demand to improve the technological facilitation of creative practices. All previous ubimus publications have featured at least one paper on this subject and the present proceedings volume is no exception. Koszolsko's (2023) *iubar project* and Barros and coauthors's (2023) *Engenhoca* provide

[2] We are not claiming that all caveats implicit in enabling programming-based techniques for non-technical participants have been addressed. This is a promising thread of research that will need to gather evidence through deployments in various settings and cultural contexts.

complementary and original perspectives on ubimus creativity. These deployments are complemented by a new conceptual approach led by Messina and developed by NAP research group. Musical stuff is defined as a phenomenology of pliable entities that enable distributed creative activities, deployable on the musical internet. Although originally motivated as a response to the decontextualising terminology of "thing" (adopted by many practitioners and researchers that employ IoT-based resources), a conceptualisation of contextually bound musical "stuff" is proposed here as a self-sufficient conceptual building-block for ubimus design, c.f. Messina et al. (2023; this volume) who have also connected this concept with the need for scalable deployments to assess the limits and potentials of emerging ubimus frameworks. An expansion of our understanding of creativity, agency, and inclusivity are themes taken up by this year's invited lecture by Prof. Andrew Hugill (University of Leicester) on the subject of Aural Diversity, i.e. approaches to creative sonic practices which do not impose a normative model of "hearing health". This invited lecture generated lively discussions in the paper sessions that followed.

Beyond the formal paper presentations, UbiMus 2023 featured, for the first time, an extensive workshop and artworks/music programme, including a ubimus intervention involving colleagues from Africa (Kenya-based Nyokabi Kariũki).[3] As part of the artistic programme, Stephen Roddy presented his *Signal to Noise Loops v5*, a data-driven audio-visual installation created using data from noise level monitors in Dublin during and after the COVID-19 pandemic, foregrounding the discernible shifts between networked communication and offline/"real-world" activities throughout this period.[4] Patrick Moore's *Treatise* created an elusive installation in which listener proximity to a headphone installation gave rise to an audible retreat. Both artworks reflect an expanded notion of agency within diverse and technologically mediated environments. The music programme (presented in a concert including multichannel immersive works) also exemplified experiences of and situation within diverse sonic environments. Iain McCurdy's *Timios Stavros* provided a particularly vivid rendering of a soundscape: at once relatively unmediated (eschewing significant processing) and yet strikingly immersive; that of a hilltop church in Western Crete, captured using the double-mid-side technique. Declan Tuite's *Ciúnas: antiphonal to ambisonic* took what was almost an opposite approach, playing studio-created materials 'through' various sites of architectural and acoustic significance, creating a dynamic ambisonic arrangement by means of site-oriented combinations and juxtapositions, yielding a creative mixed-reality perspective on sound. These ubimus-oriented ontologies of sonic perception can also be seen as operating within Shane Byrne's *Mortal Coil*, which uses inaudible electromagnetic fields (EMFs) as its artistic materials and Berk Yagli's sonic metaphor of polarising news media, *Ideological Distortion*. These musical experiences call for specific approaches. On the one side they take into account the expanded possibilities of current technological infrastructure, and on the other side they rely on local knowledge built through situated methods. We expect that perspectives from the digital humanities will help us to expand our understanding of the creative processes and products of these emergent ubimus practices.

The present publication is complemented by several upcoming special issues of journals whose topics have been motivated by discussions carried out by our community during

[3] See (Kariũki and Hofmann, 2023; this volume) for their workshop on body percussion and live electronics.

[4] This trend has been highlighted by ubimus conceptual and practical developments that since April 2020 has been labelled as *post-2020 creative practice* (Keller et al. 2020).

previous symposia. 'Ecologically Grounded Creative Practices and Ubiquitous Music, Interaction and Environment' (*Organised Sound*) and 'Ubimus contributions to digital creative practices' (*Digital Creativit*y) will provide further opportunities to develop some of the topics addressed in this editorial. More specifically, the former volume features state-of-the-art artistic projects that are grounded on a fairly well-established thread of applications of E4-cognition[5] in music. The latter collection of articles highlights developments in ubimus practice featuring design and usage of technological resources that are not aligned with the acoustic-instrumental approach. Examples of expanded materials include the use of e-textiles, the incorporation of IoT resources, and (as mentioned above) the exploration of gastrosonics. The former collection suggests the emergence of two trends within the ubimus field: techniques that are both generally applicable and culturally blind, that may tend to increase the homogeneity of the cultural products; and techniques that may be more restricted but that remain sensitive to specific cultural traits. According to Bridges, Lazzarini and Keller (2023), the latter may help to incorporate local or community-specific knowledge.[6] Both, the Digital Creativity and the Organised Sound volumes, provide excellent complements to the investigations documented in these proceedings.

For the present, this brief survey of the written contributions to the UbiMus 2023 Proceedings (as well as the symposium's invited presentations and selected artistic proposals) reveals an emerging alignment both in theory and practice among ubimus initiatives. New threads of research − exemplified by the proposals that explore cross-modal aspects of musical experiences, strategies for powerful audio-oriented prototyping, and the exploration of ways to deploy technology in non-standard settings − underline the openness of ubimus as a collection of frameworks and set of practices. This potential to integrate diverse procedures and techniques indicates the emergence of a scaffold that may foster future ubimus intersections with other fields. Considering ubimus in the form of a loose movement that benefits from consensus and shared goals, embodying increasingly diverse and distributed practices, may allow our community to address key challenges. Exploring the implications of new designs, technologies, and creative possibilities, considering these issues in the context of specific communities, perspectives, and musical experiences may yield a renewed impetus to our efforts as we consider what may constitute *third-wave ubimus*.

References to other publications:

Cullimore, J. and Gerhard, D., 2015. The Virtuoso Composer and the Formidable Machine: A Path to Preserving Human Compositional Expression. In *Proceedings of the 12th Sound and Music Computing Conference (SMC 2015)*.

Keller, D., Alcвntara-Silva, T. R. and Mesz, B. A. 2023. Editorial: Caminhos investigativos da mъsica ubнqua, gastrossфnica e bem-estar. *Revista Vórtex* 11(1), 1−33. (DOI: 10.33871/23179937.2023.11.1.7776)

Keller, D., Costalonga, L. and Messina, M. 2020. Editorial: Ubiquitous Music Making in COVID-19 Times. In *Proceedings of the Ubiquitous Music Workshop (UbiMus 2020)* (pp. 3-16). Porto Seguro, BA: Ubiquitous Music Group.

[5] E4: embedded, embodied, enactive, ecological.

[6] See the editorial of the volume Ecologically Grounded Creative Practices and Ubiquitous Music, Interaction and Environment.

Keller, D. and Lazzarini, V., 2017. Ecologically grounded creative practices in ubiquitous music. *Organised Sound*, 22(1), 61-72.

Lima, M. H., Keller, D., Pimenta, M. S., Lazzarini, V. and Miletto, E. M. (2012). Creativity-centred design for ubiquitous musical activities: Two case studies. *Journal of Music, Technology and Education* 5(2), 195-222.

Otero, N., Jansen, M., Lazzarini, V. and Keller, D., 2020. Computational thinking in ubiquitous music ecologies. In: Lazzarini, V., Keller, D., Otero, N., and Turchet, L. (eds.). *Ubiquitous Music Ecologies* (pp. 146-170). Oxford and New York: Routledge.

Contents

Full Paper Presentations

Paper Session

Ubimus Rapid Prototyping and Live Environments

Aurora-Lattice: Rapid Prototyping and Development of Music Processing Applications

Victor Lazzarini[1] and Rory Walsh[2]

[1]Music Department, Maynooth University, Maynooth, Ireland
[2]Dundalk Institute of Technology, Dundalk, Ireland

Victor.Lazzarini@mu.ie, rorywalsh@ear.ie

Abstract. *This paper describes two programming systems designed for rapid prototyping and development of sound processing applications. The first one of these is Aurora, a lightweight dependency-free header-only C++ template class library. Aurora supports a variety of techniques of musical signal processing applications from synthesis to time- and frequency-domain methods. The second system is Lattice, a graphical patcher and user interface designer for plugin and standalone application development. Lattice is an open system providing a dependency-free C++ framework for modular extensions that allows users to create efficient musical signal processing applications within a graphical envi-ronment. These two systems can combined as a toolkit and a framework, making up a comprehensive set of tools to support a rapid prototyping and development approach.*

1. Introduction

Since the beginnings of computer music, sound synthesis and processing systems have taken advantage of the object-oriented programming paradigm. This can be observed as early as 1961 in MUSIC III, which, although being not much more than a collection of macros, introduced the idea of the unit generator and the computer music instrument [Lazzarini 2013]. Both of these are classes: unit generator objects are used by composi-tion to define instruments, and these are instantiated to generate a sound waveform. The operation of such a system could be described as the setting up and running of a simula-tion. These fundamental principles were then adopted in many systems that followed, in various forms from text and graphical domain-specific languages to digital audio work-station (DAW) plugin systems.

If we take the latter as a case in hand, we can observe that, from a high-level point of view, a plugin running in a DAW is an instance of a sound synthesis or processing class. This is possibly made up of components themselves made from other well-defined classes (typical examples would be envelopes, oscillators, filters, mixers). Additionally, many plugin systems rely on well-defined frameworks, which provide an operational object-oriented interface for the set-up and running of sound processing objects. Again, we can describe a host DAW as an environment to run simulations of waveform generation processes, similarly to the runtime engines of domain-specific languages.

In this context, the C++ language offers good support and facilities for the development of components for both DAW and music programming systems. In

fact, various audio processing libraries have appeared along the years, providing common components to support this. Two of the pioneering systems were the Synthesis Toolkit (STK) [Cook and Scavone 1999] and the Sound Object Library (SndObj) [Lazzarini and Accorsi 1998, Lazzarini 2000]. They also represented two kinds of object-oriented libraries, toolkits and frameworks [Pope 1989], respectively. The former was designed to support use by composition, where objects of the class library are used directly to make up instruments, whereas the latter provided a set of classes of different kinds that could be specialised to implement certain tasks.

In some situations, these two forms of object-oriented programming may be supported; in fact, SndObj allowed extensive use as a toolkit. However, more typical is the case that a framework may be designed simply to provide a common interface: for example, the Virtual Studio Technology (VST) [Steinberg 2023], as well as similar systems, is a framework that offers nothing but a means to load, run, and control a binary module inside a host. It is very common for music processing systems to provide this type of fundamental framework for the addition of new components as part of their application programming interface (API). Csound for example has these in both C and C++ flavours [Lazzarini et al. 2016].

Equally, some libraries are strictly designed to be used as toolkits: the standard C++ library is one such example, where subclassing is not permitted. In this case, support is given to composition and users are supposed to employ library objects directly in their programs, or in classes of their own. Clearly, combining such libraries with a well-defined framework is an optimal way to employ the object-oriented paradigm. In this paper, we will describe the development of a lightweight toolkit, Aurora, designed to support the development of sound synthesis and processing instruments, and Lattice, an application framework that allows rapid prototyping of plugins and standalone programs for various platforms.

2. Aurora

Aurora provides a set of sound processing classes that can be composed to make instrument classes. It is designed to be

- lightweight and minimal,
- header-only,
- generic, and
- simple to deploy.

The library implements all essential operations in one form only and the code is reduced to be as small as possible through extensive re-use. No external dependencies beyond the standard C++ library are employed; to use it, linking is not necessary: classes are available simply by including the relevant header files. Through templating, code is provided in generic forms that may be customised by supplying user-defined functions either at compile time or through lambda constructs. Processing objects may be thought as unit generators, which produce and/or consume time-domain signals passed around as vectors. Synthesis graphs may be easily programmed by users familiar with high-level computer music languages, with only a limited amount of C++ literacy required.

2.1. Base Class and Hierarchy

Aurora has a very shallow class hierarchy. For time-domain signal processing, a base class is provided from which most classes are derived at a single level, with few exceptions. This base is effectively only a very thin wrapper around a vector, which holds the object output. Its interface only contains methods to access and modify attributes related to this vector. Internally, it provides a generic means to iterate over it and process audio samples.

Therefore, a fundamental principle of Aurora is that each object contains, as part of its state, its own output, which is provided in a read-only form to the outside world. It can only be modified by the object itself, when processing is invoked. Amongst other things, this allows for a straightforward implementation of asynchronous and multi-threading operations (e.g. in the case of parallel processing). Vector sizes can also be modified with little overhead by preallocation of memory.

Although objects of `SndBase` (the base class) can be instantiated, they are barely functional. There is no processing method defined in this class, so the output cannot be modified in any way. It is only meant to provide the most basic services that are common to all classes in the library and so, in keeping with the design principles stated above, the base class does nothing. `SndBase` has a single template argument, the sample type, which should be a floating-point type of the required precision.

Functionality is given by the derived classes. These provide generic forms of common digital signal processing operations. Each class exposes an interface consisting of one or more overloads of the functional operator. This is the outside-facing processing method, which is invoked to update the object output, always returning a read-only reference to it. This design leads to the following C++ code pattern

```
// obj of a given Aurora class using sample type Type
Aurora::Class<Type> obj;
...
// functional operator takes an input, processes it
// and returns a read-only reference to its output
auto &out = obj(in);
```

The number and types of input arguments will depend on the what process is implemented.

The most basic and generic forms of processing are implemented in the `BinOp` class. This simply applies a user-defined function to combine two inputs, of either scalar or vectorial forms, to form an output. Both the sample type and a binary function taking two arguments and producing an output of the same type are given as template parameters to it. This class exemplifies the generic aspect of the library. It can be used in applications such as mixing, offsetting, scaling, modulating, etc, depending on the function supplied. Also for signal mixing, a dedicated `Mix` class is provided, whose functional operator takes any number of input vectors. This is implemented via recursive template functions, which are all resolved into efficient code at compile time. Another basic utility class is `Buff`, which provides a simple circular buffer.

In addition to these, the following processing classes are available in Aurora:

- Oscillators: `Osc` (generic oscillator), `BlOsc` (specialist bandlimited oscillator).

- Filters: `OnePole` (first-order lowpass), `TwoPole` (state variable), `FourPole` (fourth-order resonating lowpass), `Eq` (equaliser), `Fil` (generic second-order).
- Delay: `Del` (generic delay-based operations).
- Envelope: `Env` (generic gated envelope with release).
- Function mapping: `Func` (generic function application).
- Convolution: `Conv` (partitioned convolution).

With the functionality provided by these classes in a very generic way, most of the standard time-domain processing required to construct computer music instruments can be implemented. In addition to these classes, others are provided to support particular operations:

- `FFT`: radix-2 fast Fourier transform.
- `IR`: frequency response tables for partitioned convolution.
- `TableSet`: wave tables for bandlimited oscillators.

2.2. Generic Programming

The use of generic programming in Aurora may be further exemplified in the design of the `Del`. Effectively, this is another wrapper over a vector, which implements a delay line and provides the means of accessing it circularly. The actual reading, as well as the updating of the delay is implemented via user-supplied functions, which allow for the full range of delay-based operations, from fixed and variable-time delays, comb and allpass filters, finite impulse response filters (direct convolution), to waveguides and physical models. Functions for interpolated and truncated reading from the delay line are available. Such functions can be also used with table lookup oscillators, which can be implemented using the `Osc` class, and function table applications (`Func`). These give other examples of generic programming, alongside `Env`, and `Fil`.

2.3. Examples

Two examples are provided to demonstrate the use of Aurora. The first one of these is of a synthesis instrument, and the second of a standard effect. A well-known design of frequency modulation synthesis is given by the arrangement of oscillators and envelopes in units called operators. With these, it is then possible to structure different synthesis algorithms, by combining such operators in different ways.

With Aurora, it is fairly simple to design a class that can implement an operator of this kind. All we need is a sine wave oscillator and an envelope. The class then needs to implement two functional overloads: one for the case where the phase of the oscillator is not modulated, and another where it is. These then allow us to connect operator objects in various configurations.

```
#include "Osc.h"
#include "Env.h"
using namespace Aurora;
template <typename S> struct Opm {
  static S mult(S a, S b) { return a * b; }
  Osc<S, sin<S>> osc;
  Env<S> env;
  BinOp<S, mult> amp;
  S att, dec, sus;
```

```
  S o2pi;

  Opm(S rel, S fs = def_sr, std::size_t vsize = def_vsize)
      : osc(fs), env(att, dec, sus, rel, fs), att(0), dec(0), sus(0),
        o2pi(1 / twopi) {}

  // release time setter
  void release(S rel) { env.release(rel); }

  // no modulation
  auto &operator()(S a, S f, bool gate, std::size_t vsiz = 0) {
    if (vsiz) osc.vsize(vsiz);
    return env(osc(a, f), gate);
  }

 // phase modulation
   auto &operator()(S a, S f, const std::vector<S> &pm, bool gate) {
    return env(osc(a, f, amp(o2pi, pm)), gate);
  }
};
```

With two such objects, for example, we can implement a simple FM configuration as in

```
Opm<double> op1(0.1, sr), op2(0.1, sr);
...
auto &out = op2(amp, fr, op1(index, 2*fr, gate), gate);
```

Since envelope parameters are passed as references, these class members can be updated at any time to reflect changes in the ADS configuration. Release time is treated separately with a dedicated method. Due to the generic design of the envelope class, it is possible to replace the three-stage envelope with any number of stages by supplying a different function.

Equally straightforward is implementing audio effects. For example, a stereo chorus may be designed with two delay lines modulated by two separate oscillators. The delay lines are created by supplying a variable delay interpolated lookup function (`vdelayi`) to the class template. The oscillators produce cosine waves (which is given by their default template parameter). We can compute first the two separate channels of a dual chorus.

To make this into a stereo chorus effect, the output of each delay can the combined with the input signal and the other delay to make up the left and right channel signals. We can add a width control between 0 - 1, to define the chorus stereo separation. In this example, `in` is the mono input signal vector, which is fed to each channel and `output`, the stereo output, which is encapsulated by the stereo chorus object.

```
#include "Osc.h"
#include "Del.h"
struct StereoChorus {
  static float add(float a, float b) { return a + b; }
  std::vector<float> out;
  std::array<Osc<float>, 2> lfo;
  std::array<Del<float, vdelayi>, 2> delay;
  BinOp<float, add> offs;
```

```
  StereoChorus(float sr, std::size_t vsize = def_vsize)
    :    out(vsize*2),
         lfo{Osc<float>(sr, vsize), Osc<float>(sr, vsize)},
         delay{Del<float, vdelayi>(0.1, sr, vsize),
               Del<float, vdelayi>(0.1, sr, vsize)},
         offs(vsize){};

  // one channel of dual chorus
  auto &chorus(const std::vector<float> &in, float fr, float d, int chn) {
    lfo[chn].vsize(in.size());
    return delay[chn](in, offs(d, lfo[chn](d * 0.1, fr)), 0, 1);
  }

 // stereo chorus
  auto &operator()(const std::vector<float> &in, float width) {
          bool chn = 0;
          std::size_t n = 0;
          out.resize(in.size()*2);
          auto &l = chorus(in, .93f, .017f, 0);
          auto &r = chorus(in, .87f, .013f, 1);
          for (auto &s : out) {
            s = (chn = !chn) ? l[n] * width - r[n] * (1-width)
                             : r[n] * width - l[n] * (1-width);
            n = !chn ? n + 1 : n;
          }
          return out;
  }
};
```

With an object of this class, we can turn a mono signal into a stereo output with a given width and gentle modulation,

```
StereoChorus chorus(sr);
...
auto &out = chorus(in, 0.7f);
```

3. Lattice

Lattice is a system for quickly prototyping and developing audio plugins. At its core is a host application (AU / VST3 / Standalone), which contains a digital audio graph. The graph can host any number of custom processing modules. A C++ framework for developing modules is provided, without any external dependencies. Although any processing library can be used, the lightweight nature of Aurora makes it a perfect fit for such modules. Each processing module, compiled as a dynamic library, when loaded in the graph is given a generic editor which sound designers can use to quickly access each module's underlying parameters. Plugin user interfaces are created using a simple web based frontend. A JSON editor, accessible in the graph, lets users declare what parameters from the graph they wish to expose on their plugin's main UI. A simple generic interface is provided out of the box, with sliders, buttons, etc., but the system is open-ended enough to allow the development of any custom controls and UI through simple JS/HTML/CSS.

Lattice also exists as an embeddable graph library, that is, the entire graph and processing modules can be loaded into any application through a very simple and intuitive interface. Users can load any high-level graph developed in the main Lattice host

application. The graph engine, and all module parameters, can be accessed in the derived application. Users can also mix the Lattice graph engine with any other processing interfaces. Although Lattice was written using the JUCE audio framework, the module API and embeddable graph library have does not depend on that framework. Development of modules and plugins with Lattice has no external dependencies.

3.1. The Lattice module API

The API is tied closely to common plugin formats. The `LatticeProcessorModule` base class can be used to develop both synths and audio effects. The main difference between these two types of objects is that the former can be instantiated multiple times (one for each synthesiser voice) and the latter is only instantiated once; that is, when a synth voice is created in the Lattice patcher it can actually spawn more than one instance of the module (depending on the polyphony settings defined for it).

As a basic wrapper for audio plugins, The `LatticeProcessorModule` provides overridable methods for following tasks,

- Declaring channel layouts: the `createChannels()` method returns a `ChannelData` structure which holds a list of `std::vector` of `Channel` objects. Each channel is marked as either input or output. Users can push as many input and output channels as they wish to this structure.
- Declaring a list of module parameters: the `createParameters()` returns a `ParameterData` structure, which holds a list of `Parameter` objects. The `Parameter` class contains the following members:
  - Name: the parameter name as it appears in the Lattice graph
  - Range: a struct to hold the min, max, default value, increment size, and skew values for the parameter
  - Type: determines the parameter type to be displayed when the users open a module node in the graph. Type can be a trigger, switch, momentary button, slider, or a file button.
- Accessing parameter changes: changes to module parameters in the Lattice graph trigger the
  `hostParameterChanged(const char* parameterId, float newValue)`
  callback function. The `parameterID` is a combination of the unique name for the module, assigned by the host, and the parameter name itself, i.e, “Super Synth 11 - Attack”. The `getParameterName()` method can be used to extract the parameter name. In many cases we can just call `getParameter()` directly from the processing method.
- Updating a host from a module: calling
  `updateHostParameter(const char* parameterID, float newValue)`
  will notify the host that a parameter needs to be updated.
- Accessing basic MIDI information: there are several utility functions that can be overridden to access information about incoming MIDI notes. `startNote()` and `stopNote()` will be triggered whenever a user presses or releases a MIDI note. There are also functions to notify changes to MIDI program and pitch bend.
- Processing: the core functionality of an application is its audio processing method. The module API provides the following overridable methods, depending on the type of module,

- `process(float** buffer, std::size_t blockSize)`: this is used by processing modules taking in non-interleaved data from `buffer` and replacing it with processed outputs. Each channel is an audio vector with `blockSize` samples.
- `processSynthVoice(float** buffer, std::size_t blockSize)`: this is used by modules implementing synthesiser voices; each voice places its output data in `buffer`.
  `processSamplerVoice(float** buffer, std::size_t blockSize)`: is a special version of the above for implementing sampling synthesiser voices.
- For MIDI-processing modules we should use

```
processMidi(std::size_t blockSize,
            std::vector<LatticeMidiMessage>& midiMessages))
```

Each module is compiled into a dynamic library that can be loaded in the Lattice host and used for the rapid development of audio applications. Users can combine as many modules as they like into a high level processing graph.

3.2. The Lattice host application

The Lattice host application is where the audio graph is put together out of the available modules. When the application is started, the patcher window is shown containing the input and output modules as default (audio and MIDI), as well as a virtual MIDI keyboard for testing. Patching input to output makes a loopback connection (Fig. 1). All modules located in the default Lattice module directory will be listed in the patcher drop-down context menu. Instances of these can be created and then connected together to define the control and audio signal flow for the application or plugin. Modules are listed according to their categories.

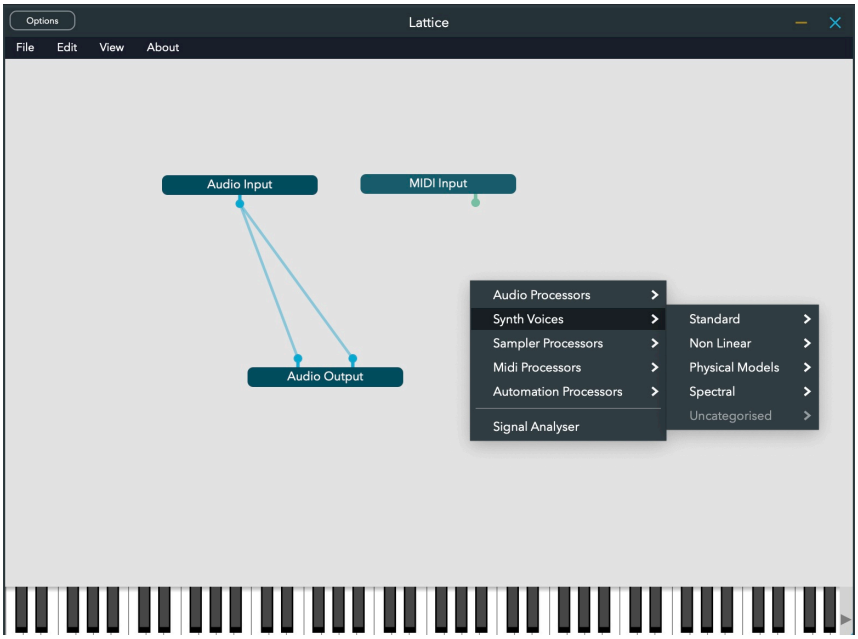


Figure 1. The Lattice patcher.

For example, to create a synth voice, we can select a module from the relevant list, connect the MIDI input to it, and patch its output to the audio output object. If we want to set the module parameters, double clicking on the object shows its parameters (Fig. 2). Depending on the module, parameters may be exposed to be automated, that is controlled by an external object. To do that we can bring up the object context menu (right click) and select the required parameter. An input pin will appear which can be used to connect the output of an automation controller (such as a low frequency oscillator).

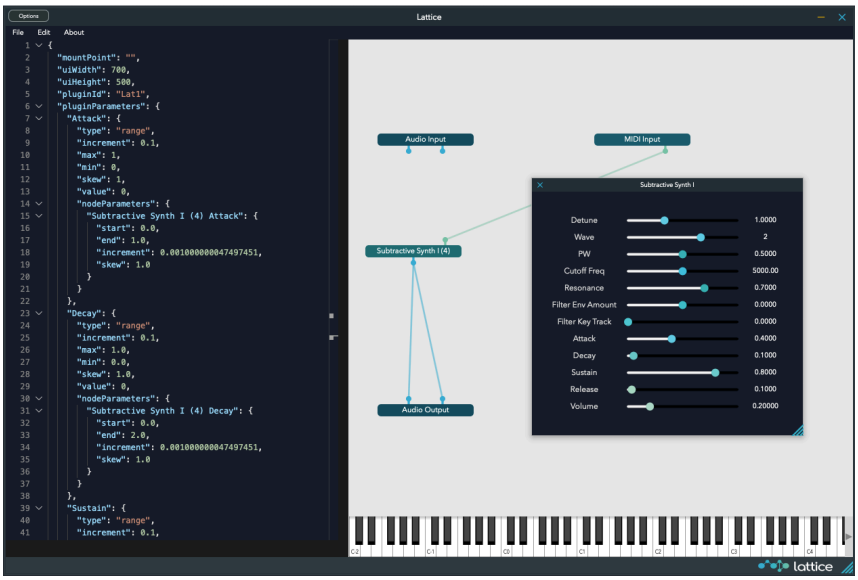


Figure 2. A simple patch and object parameter dialog window.

Once a patch is finalised, we can start to build its user interface by defining what parameters to expose. To do so, we can bring up the built-in JSON properties editor. It is here that we can define things like the plugin size, the plugin ID, and most importantly, what parameters to expose in the main UI. Widgets such as sliders, buttons, etc., are defined in the `pluginParameters` object. For each widget, one must also fill the `nodeParameters` object with a parameter, or parameters, to which this widget will be attached. You can attach a single widget to as many graph parameters as you wish, and can also create user-defined mapping. Auto-complete functionality makes it easy for the user to edit and modify the JSON properties. An example of a generic UI for the simple patch shown earlier (Fig. 2) is depicted in Fig. 3)

Each Lattice project must be given a default mount point, which must contain an index.html. It is this page that Lattice will load in its webview once you switch to plugin mode. When the page is loaded, the JSON properties file will be sent to the web frontend for parsing through a 'latticeOnLoad' event. Events can then be sent between the web frontend and Lattice to notify of parameter updates, preset changes, incoming MIDI notes, etc. The real power behind the web based UI is that users can develop custom frontends using any number of already established JS frameworks. The generic frontend that ships with Lattice was developed with Svelte, but it is also possible to use React, Vue, Angular, etc. Users can also define custom JSON objects and higher level templates making it possible to develop their own in-house UI system that can be created entirely through the Lattice JSON properties file.

Figure 3. A generic interface.

4. Aurora and Lattice

Aurora has been integrated into Lattice to provide the fundamental modules that are offered in the application. Currently there are 62 modules provided with the system, most of these have been written using template classes from the Aurora library. The range of modules include various complete types of synthesiser voices, sampling, audio effects, automation, MIDI processors, as well as visualisation support.

Lattice modules are not unit generators as found in music programming systems. They are designed at a higher level of functionality, and typically are constructed out of such components (like oscillators, filters, envelopes) connected together to give a complete patch element. This has two main advantages: since the elements are connected inside the compiled module, each module is more efficient than a patch of unit generators thrown together in a system like Pure Data [Puckette 2023]; they are also easier to use to design plugins based on the combination of typical signal processing operations, supporting a rapid development approach.

However, Lattice is an open system. It is possible to create modules that are written as unit generators by selecting the relevant class in Aurora, or writing the C or C++ code directly, and wrapping it using the module API. The system is designed so that users may provided any modules as they wish, and use the Lattice patcher as a means of constructing their instruments from lower-level components. However, we have elected to provide pre-compiled modules at a higher level to support the creative work of plugin designers who may prefer to work with more complete units in the patcher. In fact, this is a feature seldom seen in music programming systems, which tend to offer a finer granularity in terms of synthesis components. This is something that sometimes may pose a higher

barrier to adoption.

5. Conclusions

In this paper, we have described two systems for the rapid prototyping and development of musical signal processing applications. The first of these is a C++ class template library, Aurora, which provides the means for putting together various kinds of processes for sound transformation. It includes support for both time and frequency domain processing, and it has no dependencies beyond the C++ standard library. Code examples of synthesis and processing applications were shown. The library is licensed under the BSD 3-clause license, which allows permissive use of the source code. Aurora, together with a number of examples and reference documentation, can be downloaded from its repository at

`https://github.com/vlazzarini/aurora`

The second system discussed in this paper is Lattice, a graphic environment for music application development complete with a C++ framework. This provides two essential components: a patcher and a web-based front end for the development of custom UIs, which together with its open-source module API allow users to design plugins and standalone applications from the ground up. Combined with Aurora, which is used to provide the code for module development, it provides strong support for rapid prototyping and development of musical applications. The Lattice module SDK can be found at

`https://github.com/rorywalsh/lattice_modules`

References

Cook, P. and Scavone, G. (1999). The synthesis toolkit (stk). In *Proceedings of the ICMC 99*, volume III, pages 164–166, Berlin.

Lazzarini, V. (2000). The SndObj sound object library. *Organised Sound*, 2(5):35–49.

Lazzarini, V. (2013). The development of computer music programming systems. *Journal of New Music Research*, 42(1):97–110.

Lazzarini, V. and Accorsi, F. (1998). Designing a sound object library. In *Proceedings of the XVIII Brazilian Computer Society Conference*, volume III, pages 95–104, Belo Horizonte.

Lazzarini, V., Yi, S., Ffitch, J., Heintz, J., Brandtsegg, O., and McCurdy, I. (2016). *Csound: A Sound and Music Computing System*. Springer, Berlin, 1st edition.

Pope, S. T. (1989). Machine tongues xi: Object-oriented software design. *Computer Music Journal*, 13(2):9–22.

Puckette, M. (2023). Pure data. `https://puredata.info/`.

Steinberg (2023). Virtual studio technology. `https://steinbergmedia.github.io/vst3_doc/vstsdk/index.html`.

Radial String Chimes: A Behavioural Driven Tangible Instrument Derived from Networked Music Performance Practices

Álvaro Barbosa[1], Gérald Estadieu[1], Ng Ka Man[1]

[1]Faculty of Arts and Humanities – University of Saint Joseph
Macau S.A.R. - China

{abarbosa,gestadieu,sandra.ng}@usj.edu.mo

Abstract. *Research in ubiquitous networked music systems has unveiled the potential of behavioural-driven interaction interfaces as an effective model to cope with network communication delays in remote musical performances. Most of the techniques developed under these premises are based on digital music interfaces implemented on laptop computers or tablet devices, where a certain degree of gestural control comes as an added dimension. The purpose of this paper is to present an implementation of such type of interfaces in the form of a physical tangible musical instrument, contemplating multiple expressive possibilities. This is viable at the current stage of technological development thanks to leveraging 3D printing and laser cutting technologies for effective prototyping and testing of such a device. The paper seeks to demonstrate that this approach opens a wide range of possibilities for creating musical instruments with versatility and expressiveness beyond what is usually accomplished in traditional instruments. This implementation, entitled "Radial String Chimes," is presented with its advantages, the challenges it faces, and the methods used to create it. Finally, the paper will offer suggestions for further developing such an instrument to unlock its potential.*

1. Introduction

The Radial String Chimes is an artistic research project that aims to implement an interactive musical interface that behaves as an instrument to experiment with musical creativity by introducing the principles of behaviour-driven interaction to control the sound generated by acoustic strings.

1.1 From Networked Music to Behaviour-Driven Interaction

Networked Music is an established discipline with initial significant developments since the late 90's and first surveyed substantially in (Barbosa, 2006). Networked Music encapsulates the challenges of musical collaboration over the internet, integrating this medium's acoustic properties in the musical creativity and production processes. The most distinctive feature of the Networked Music acoustic space, particularly over long distances, is its high latency and consequent acoustic lag created between performers (Barbosa & Cordeiro, 2011). On the other hand, Interaction between Humans and Computers is a crucial topic in all disciplines, but particularly in music performance, it takes a central role in the process of designing new musical instruments. Early work by Marc Battier and Marcelo Wanderley (Wanderley & Battier, 2000) introduced this topic in the domain with systematic academic research, and multiple

models and frameworks have been derived since then. In particular, considering the challenges of the interaction between humans and computer-based music generation in the Networked Music field, the mentioned prevalence of high network latencies between collaborative participants on joint performances and the latency between the sound engine and interfaces can be critical, as presented in (Barbosa et al., 2003).

One way to address this impediment to thigh real-time synchronous performance is the development of semi-generative interfaces for software musical engines that introduced a loose coupling interaction/performance model (like in an extreme case the action/response interaction between Mission Control on Earth and the Mars Rover) in opposition to a thigh coupling model (like you have in a real-time acoustic instrument, such as a piano). This kind of interaction is much more immune to latency disruption since the system's response time is much higher than the network by design. An early example of this kind of system is the Public Sound Objects (Barbosa, Cardoso & Geiger, 2005), and many recent systems have emerged in the last decade in relation to Networked Music Performances, Ubiquitous Music, and the Internet of Musical Things (Turchet & Rottondi, 2022)

In these cases where extreme delay exists between the generation of sound (or visuals) and the gesture that triggers this event, it is common to resort to solutions where the synthesis is generated in an algorithmic way performing an ongoing behaviour in a way that the interaction will trigger changes in this behaviour, but it will not have a perceivable instant effect while maintaining a certain degree of control to the performer.

1.3 Original Radial String Chimes

Deriving from the attempt to address latency challenges in Networked Music, Behaviour Driven Interaction has been common in digital performance as a way to allow the performers to have a higher-level control over the performative events and has mostly been applied in digital music Instruments. In the same way, the Radial String Chimes Project, presented in this paper, it is a system with similar characteristics in the form of a tangible acoustic musical instrument. It has generative behaviours in a tangible form that is loosely coupled with the gestural physical interaction of the performer.

In 2010, Álvaro Barbosa designed and prototyped the first Radial String Chimes instrument (Barbosa, 2011) as part of his research on networked music at the Center for Computer Research in Music and Acoustics from Stanford University (CCRMA). It is defined as an interactive musical device triggered by motion applied to a spinning vinyl record. The musician can spin the record, making hanging coffee straws to bounce and pluck twelve radial guitar strings applied to a round wooden board. Using an ebow[1] on a string, one can also get it into resonance and generate a different sound effect. These musical sounds can further be expanded in its expressive range by coupling it with additional digital signal processing units.

2. Towards an Open Design

We approached this research and experimental work with a design perspective and as such, we used the Design Thinking Methodology (Wolniak, 2017). Indeed, Design Thinking is well-acknowledged as a novel problem-solving methodology that encourages

1 https://ebow.com/

creativity and innovation to reach a solution (Liedtka, 2018). The methodology, has originally promoted by the design company IDEO (Kelley & Littman, 2001), is composed by five stages as a lifecycle to continuously innovate in solving a problem: “empathise”, “define”, “ideate”, “prototype”, “test”.

After using the instrument for several musical events over the past decade[2], it became important to rethink the design of the instrument and its possibilities. The instrument was used as intended and served well. Still, with technological advancement, we quickly realised that we could have an opportunity to bring new features and push further its potential.

We started by watching the artist playing the instrument and discussing with him to understand the identifiable problems based on his experience - the first phase of the methodology, “empathise”. We then identified the major elements of redesign - the second phase, “define”, namely:

- The difficulty of reproducing such an instrument.
- The acoustic limitation.
- The need to communicate and share more about the instrument.
- The will to keep the same structural parts (a circular board and a circular top).

The ideation phase of the methodology ended up with a short list of five major improvements, namely:

- An acoustic body, including a soundboard with soundholes.
- An acrylic piece will replace the top vinyl disk.
- Design-specific parts will be 3D printed.
- All metallic parts will be changed to easy-to-source parts.
- The design and fabrication process will be fully documented online and open-sourced to encourage making, innovation and tinkering – hopefully creating a community.

Considering these elements, we proposed to develop a new version of the instrument that can encompass all the requirements listed above – the third phase, “ideate”. The modifications of the Radial String Chimes gave us an initial prototype – the fourth phase, “prototype”, to test with the features described in the following section below.

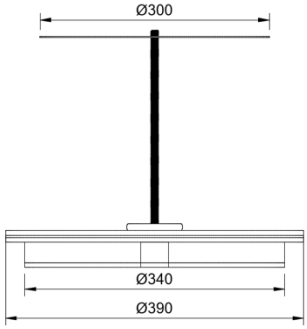


Figure 1. Technical Drawing – Simplified Front view

2 http://www.icm.gov.mo/fam/27/en/event.aspx?oldID=2567

3. Design and Technical Specifications

The five improvements mentioned above guided our design, prototyping, and testing phases. We are presenting the result of our design process in the sections below.

3.1 Acoustic Body

As guitar steel strings were attached to a wooden board in the initial version, the acoustic sounds generated, without the amplification of the piezo sensor, were limited in resonance and strength. The first step was to propose an acoustic body similar to traditional musical instruments such as violin or guitar (French, 2012). This traditional craftsmanship allows strings to resonate and sounds to get the possibility to bounce on wood and get their own tonalities (Caldersmith, 1995). The first version of the Radial String Chimes used a circular shape for the wooden board. We decided to keep this feature as it allows the instrument to be positioned anywhere, and we only have to consider the strings' position to play. It is an efficient and simple design decision that simplifies the musician's behaviour with the instrument.

The original design used a standard kitchen wooden board of 30 cm in diameter which could fit up to two guitar string sets (twelve strings total) with their corresponding tuning keys. As we proposed a new customised acoustic body, we enlarged it to a 39 cm diameter (Fig 1). It allowed us to add an extra string set (up to 18 strings).

Since we added an acoustic body, the second step focused on the soundboard - the top of the body. Contrary to the original board, we had to add soundholes to allow projection of the sound, as applied in a guitar (Elejabarrieta et al., 2002). In the first prototype, we decided to go with two F-Holes (Fig. 2) - sound holes found on traditional violins. However, as we will explain later in this paper, the open-design approach allows anyone to choose their own sound hole design - of course, sound might be affected too.

Finally, we added two piezo sensors to the soundboard's inner part as part of the acoustic body. These two piezo are connected to a stereo 6.35mm audio jack located on the side of the acoustic body. As expected, this connection allows for further experimentation and sound manipulation via dedicated effects systems or computers.

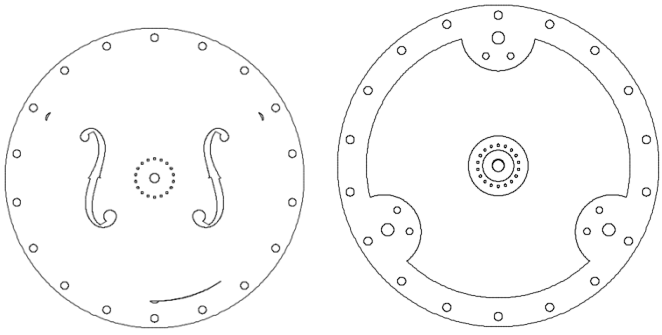


Figure 2. Technical Drawing – Simplified Top view of the Acoustic Body

3.2 Fabrication Process - DIY

The Maker Movement (Dougherty et al., 2016; Gershenfeld, 2012), in its global approach and, in particular, its hacking philosophy, has an impact in various physical or digital areas such as Education (Halverson & Sheridan, 2014), Fabrication (Varotto et al., 2017)

but also to a larger scale Museology (Estadieu & Caires, 2018; Rey, 2017), city space (Kozsilovics, 2016) or Business and Marketing with the concept of Growth Hacking (Conway & Hemphill, 2019). It is not surprising to find these concepts applied to Arts and particularly to musical instruments (Timoney et al., 2020), whether it is traditional instruments augmented (Bernardini et al., 2013; Bryan-Kinns & Li, 2020; Juniawan et al., 2020) or the creation of a new instrument (Brown & Ferguson, 2021).

One of the decisions from the ideation phase was to use digital fabrication processes and machines. This choice has several benefits for the development of the instrument that we described below.

Figure 3. In black colour, the 3D Printed PLA parts of the instrument

1. Prototyping Cycle - The prototyping cycle corresponds to a loop of the prototyping and testing phases. As all designed parts are created in 3D CAD software, it is faster to produce a physical representation, test it and go back to the computer version to modify, adapt, or improve it.

2. Cost and Time – Producing design-specific parts can be extremely costly in a traditional production model and will usually take months, back and forth from our computer to the manufacturer.

3. DIY - As one of the objectives is to allow more musicians to use this instrument, we consider the best-practice of “Do It Yourself” (DIY) to encourage musicians alike to experiment on creating their own instrument as well as lower the cost of the instrument for musicians. We strongly believe the fabrication process to be part of the experience to understand the instrument.

4. Customisation and Tinkering - As all parts of the instruments are freely available as digital files, it empowers any musician to customise or improve their own version of the instrument and ultimately share their experiments with the community as a benefit for all.

With this approach in mind, we created the following parts:

- 4 x 3D printed parts (Fig. 3),
- 5 x CNC parts (it can be a laser cutter or a CNC router machine),

- All metallic parts are easy to find in any hardware store.

We replaced the vinyl 33 cm disk on the instrument's top with a black opaque 4mm cast acrylic to allow digital fabrication and possible customisation – vinyl is still possible, though. Finally, in the first prototype, the acoustic body comprises 9mm plywood sheets, and the soundboard is 6mm plywood (Fig. 4).

Figure 4. The Radial String Chimes v2

3.3 Open-Source Approach

We chose to license all hardware, documentation, and fabrication processes under open-source licenses to allow a DIY approach. The hardware is under the CERN-OHL-S[3] license – CERN Open Hardware License, the S is for the strongly-reciprocal variant. As for the documentation, it is under the Creative Common Attribution-ShareAlike 4.0 International – CC BY-SA 4.0[4]. All information and files related to the fabrication process are available on the dedicated website[5].

Whenever developers, makers or artists choose an open-source style license, it is always important to understand the conditions and ensure they are the most adapted to the objectives. As for this project, one of our main focuses is to give a chance to artists to build, customise and even improve the instrument; we decided to opt for a hardware license that can create a community and encourage people to share their modifications. We believe it is the best way to promote a positive creativity spiral in a community.

4. Conclusions

Expanding the principles of Behaviour-Driven Interaction to a tangible musical instrument in the form of an experimental design, opened up new perspectives of musical expressiveness and sonic originality, even at its early stage of discovery and exploration. Furthermore, applying a well-acknowledge methodology of Design Thinking, we have

[3] https://ohwr.org/cern_ohl_s_v2.pdf
[4] https://creativecommons.org/licenses/by-sa/4.0/
[5] https://usj.edu.mo/rsc

been able to identify limitations of the original musical instrument and attempt to find new possibilities by solving these shortcomings. Those limitations were of two types: 1) the physicality of the instrument itself, and 2) the user interactions and behaviour.

The prototype has been tested and used in musical performance and composition events by a limited number of artists, but developing the instrument as an open design architecture, the prospect of expanding the artist user base increases exponentially. This, by itself will create the opportunity for a much better assessment of the relevance and potential of behaviour driven interaction applied to a tangible music instrument.

As for future work, we identified 3 short-term developments. First, as we publish all files and documentation online to empower a community, we intend to open a discussion forum, and develop some video tutorials and a shop to simplify the construction of the instrument if one does not have access to a fab lab.

Second, as we have used plywood for the acoustic body prototype, we want to study the acoustic quality of wood species used in traditional lutherie – particularly guitar.

Finally, as part of our intention to study the music education potential, we will develop a series of workshops to help participants to create their own Radial String Chimes instrument and encourage them to explore both the realm of tinkering and the realm of music creation.

5. References

Barbosa, A. (2006). Computer-Supported Cooperative Work for Music Applications. PhD Thesis – Music technology Group; Pompeu Fabra University, Barcelona, Spain.

Barbosa, A., Cordeiro, J. (2011). The Influence of Perceptual Attack Times in Networked Music Performance. Proceedings of the 44th Audio Engineering Society Conference on Audio Networking; San Diego – CA, USA.

Barbosa, A., Cardoso, J., Geiger, G. (2005). Network Latency Adaptive Tempo in the Public Sound Objects System. Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2005); Vancouver, Canada.

Barbosa, A. (2011). Radial String Chimes (espanta espíritos de cordas radiais) (Instituto Nacional da Propriedade Intelectual Patent No. DP/54/2011/19771).

Barbosa, A., Kaltenbrunner, M., & Geiger, G. (2003). Interface decoupled applications for geographically displaced collaboration in music. ICMC.

Bernardini, N., Gotzen, A. de, & Vidolin, A. (2013). Augmenting traditional instruments with a motion capture system. CHI 2013 “Changing Perspectives,” Paris, France.

https://www.academia.edu/16348993/Augmenting_traditional_instruments_with_a_motion_capture_system

Brown, A., & Ferguson, J. (2021, September 6). Enhancing DIY musical instruments with custom circuit boards.

Bryan-Kinns, N., & Li, Z. (2020). ReImagining: Cross-cultural co-creation of a Chinese traditional musical instrument with digital technologies. In NIME'20, 1–6.

Caldersmith, G. (1995). Designing a guitar family. Applied Acoustics, 46(1), 3–17. https://doi.org/10.1016/0003-682X(95)93949-I

Conway, T., & Hemphill, T. (2019). Growth hacking as an approach to producing growth amongst UK technology start-ups: An evaluation. Journal of Research in Marketing and Entrepreneurship, 21(2), 163–179. https://doi.org/10.1108/JRME-12-2018-0065

Dougherty, D., O'Reilly, T., & Conrad, A. (2016). Free to Make: How the Maker Movement is Changing Our Schools, Our Jobs, and Our Minds. North Atlantic Books.

Elejabarrieta, M. J., Santamaría, C., & Ezcurra, A. (2002). Air Cavity Modes in the Resonance Box of the Guitar: The Effect of the Sound Hole. Journal of Sound and Vibration, 252(3), 584–590. https://doi.org/10.1006/jsvi.2001.3948

Estadieu, G., & Caires, C. S. (2018). Hacking: Toward a Creative Methodology for Cultural Institutions. Lisbon Consortium, Lisbon.

French, R. M. (2012). Technology of the Guitar. Springer Science & Business Media.

Gershenfeld, N. (2012). How to Make Almost Anything: The Digital Fabrication Revolution. Foreign Affairs, 91(6), 43–57.

Halverson, E. R., & Sheridan, K. (2014). The Maker Movement in Education. Harvard Educational Review, 84(4). https://doi.org/10.17763/haer.84.4.34j1g68140382063

Juniawan, F. P., Sylfania, D. Y., Pradana, H. A., & Laurentinus, L. (2020). Motivation Measurement of an Augmented Reality Traditional Musical Instruments System. Proceedings of the Sriwijaya International Conference on Information

Technology and Its Applications (SICONIAN 2019). Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019), Palembang, Indonesia. https://doi.org/10.2991/aisr.k.200424.092

Kelley, T., & Littman, J. (2001). The Art of Innovation – Lessons in Creativity from IDEO America's Leading Design Firm.

Kozsilovics, V. (2016, November 26). Le hacking urbain, quand l'art incise la ville | Observatoire de l'art contemporain. http://observatoire-art-contemporain.com/revue_decryptage/tendance_a_suivre.php?id=20120848

Liedtka, J. (2018). Why design thinking works. Harvard Business Review, 96(5), 72–79.

Rey, S. (2017). Museomix: Lessons learned from an open creative hackathon in museums. CEUR Workshop Proceedings, 1861, 24–28.

Timoney, J., Lazzarini, V., & Keller, D. (2020). DIY electronics for ubiquitous music ecosystems. In Ubiquitous Music Ecologies. Routledge.

Turchet, L., & Rottondi, C. (2022). On the relation between the fields of Networked Music Performances, Ubiquitous Music, and Internet of Musical Things. Personal and Ubiquitous Computing. https://doi.org/10.1007/s00779-022-01691-z

Varotto, C. B., Ermacora, G., & Russo, L. O. (2017). Hackability, Digital Fabrication, Technology, and Design for Social Impact. Digital Fabrication, 4.

Wanderley, M., & Battier, M. (2000). Trends in Gestural Control of Music. https://www.semanticscholar.org/paper/Trends-in-Gestural-Control-of-Music-Wanderley-Battier/bca6ad4a273d604edd8e8b6c4f80dbe1c6cdc010

Wolniak, R. (2017). The Design Thinking method and its stages. Systemy Wspomagania w Inżynierii Produkcji, 6(6), 247–255.

Toward lite coding: Designing the emugel prototype to boost music-computational thinking

Rongge Zheng[1], Damián Keller[2, 4], Joseph Timoney[1,4], Victor Lazzarini[1,4], Marcello Messina[2, 3, 4]
[1] Department of Computer Science, Maynooth University, Maynooth, Ireland
[2] NAP, Federal University of Acre, Federal University of Paraiba, Brazil
[3] Southern Federal University, Russia
[4] Ubiquitous Music Group
ronggezheng@outlook.com, dkeller@ccrma.stanford.edu

Abstract

The concept of lite coding is proposed as a place-holder for a growing body of practices within ubimus. We present the Erlang Music Generation Library 1.0 (emugel), a prototype for casual musicking that features a simplified notation to represent pitch, durations, dynamics and timbre with an emphasis on handling event patterns. We describe the design process, involving multiple cycles of implementation and deployment including specialists' analysis and usage by non-musicians. We present preliminary results gathered during the deployments. The emugel prototype has unique characteristics that set it apart from other software geared toward non-specialists. It also features good potential for power users. The results suggest that lite coding may constitute a practice well-suited for music-educational endeavours.

Introduction

Lite coding can be described as an approach to music-programming tailored for non-programmers. This is a form of lay-musician interaction in the sense that it may rely on musical knowledge without the need to exclude non-musicians (this exclusion is typical of systems based on acoustic-instrumental concepts such as common-practice musical notation). As an attempt to establish a minimum common vocabulary among practitioners, lite coding proposes a reduction of means to boost learnability and shareability. 'Light' in the sense of pliable, accessible, replicable or modular may be applied to both persistent and volatile sonic resources and it may also be extensible to other modalities, involving visual, haptic or multimodal information. The use of code as an interface may enhance its portability, opening opportunities to investigate the limits of semantics within the context of musical knowledge-sharing. Another characteristic of lite coding that sets it apart from other music-programming initiatives is its non-hierarchical or 'patchy' approach to musical information. As lite coders, we are not interested in a top-down, all-encompassing language (cf. Mathews *et al*., 1969 for a historically significant example of an acoustic-instrumental perspective on language design). This patchy approach to design has a counterpart in the adaptive-opportunistic practices that characterise various ubimus initiatives (Keller, Messina and Oliveira 2020). Multiple genealogies of lite coding can in fact be identified and traced back to ubimus endeavours. Several early ubimus works tackle everyday creativity via specialised music-making software, testing the productions, activities and reactions of subjects describable as "lay" participants[1], new to both

[1] We avoid, on the one hand, the label "user" (widely adopted in human-computer interaction) because ubimus practice is not necessarily divided between consumers and producers of technology and know-how. On the other hand, we take a careful stance regarding the lack of domain-specific training. Lay participants are enabled as

musical activity and computer programming (Messina *et al*., 2019; Messina *et al*., 2021). We argue that from these contrasting experiences emerges a consistent connection between efforts aimed at democratising access to live coding practices and a preoccupation to overcome models of musical interaction that enforce social oppression, hegemony and coercivity.[2]

Erlang Music Generation Library

To make it easier for lay participants to make music, we have designed and implemented the Erlang Music Generation Library (emugel). This library provides functions for users to create music and listen to the sonic results. The proposal incorporates current sonic-generation functionality from Erlang, Python and Sonic Pi. Thus, our library supports casual users, allowing them to express their musical ideas through the practice of lite coding.

The library features six functions to generate musical processes, organised as independent tracks. The tracks can be freely handled or can be timed concurrently, as all the tracks have a `delay` parameter to define their onset. Some of them feature a `duration` parameter to enable detailed temporal control. Moreover, each of the tracks supports a different strategy for music-making: the function `beats_per_minute` controls the tempo of periodic sequences; the function `pitch_based_track` generates pitch sequences defined as MIDI notes or alphanumeric values (i.e., pitch classes); the function `chord_play` allows to pile up long lists of pitch-based aggregates without the demand to explicitly write all the values; the function `sample_pattern_track` plays a set of samples according to a user-defined pattern; the function `rubato_pattern_track` generates events whose durations change gradually according to three parameters: initial duration, end duration and an event/no-event sequence (defined through `xo` notation); the function `choose_play` yields random sequences based on lists of pitches and durations.

To summarise, without fixed limits to their combination users have six kinds of strategies at their disposal. The sonic results can be used by mixing the tracks as synchronous temporal blocks of onsets and durations or they can be used as sonic resources for manipulation by means of third-party software (such as a DAW) to mix the independent tracks asynchronously. In other words, emugel provides support for synchronous live coding but it is especially tailored for quasi-synchronous usage.

The emugel architecture

Figure 1 describes the architecture of the emugel system. Users write scripts on a preset template file called ‘user_case.erl’ and employ the functions described in the documentation. The Main module starts the program and controls the sonic rendition. To generate sound, the Main module passes the sketch file name to the Sound_generation module. The Sound_generation module then transfers the sketch/music file to the Sonic Pi code and plays the audio. During this process, the Sonic Pi Tool is used as a bridge between Sonic Pi and

agents of their actions, not just followers of the enlightened few that possess musical training as implied by musical-interaction research focused on “performance virtuosity”.

[2] The political implications of ubimus practices are emerging as a key concern shared by several research initiatives with very diverse aims, for instance, applications in well-being, enhanced sustainability and accessibility. A precedent can be traced to the ubimus dialogical and participatory approaches to education proposed by Lima *et al*., (2012) and (2017).

emugel. The generated Sonic Pi code is stored in a cache file named ‘out.rb’. This tool reads the code from the cache file and sends it to the Sonic Pi server. Thus, we avoid the usage of the Sonic Pi GUI.

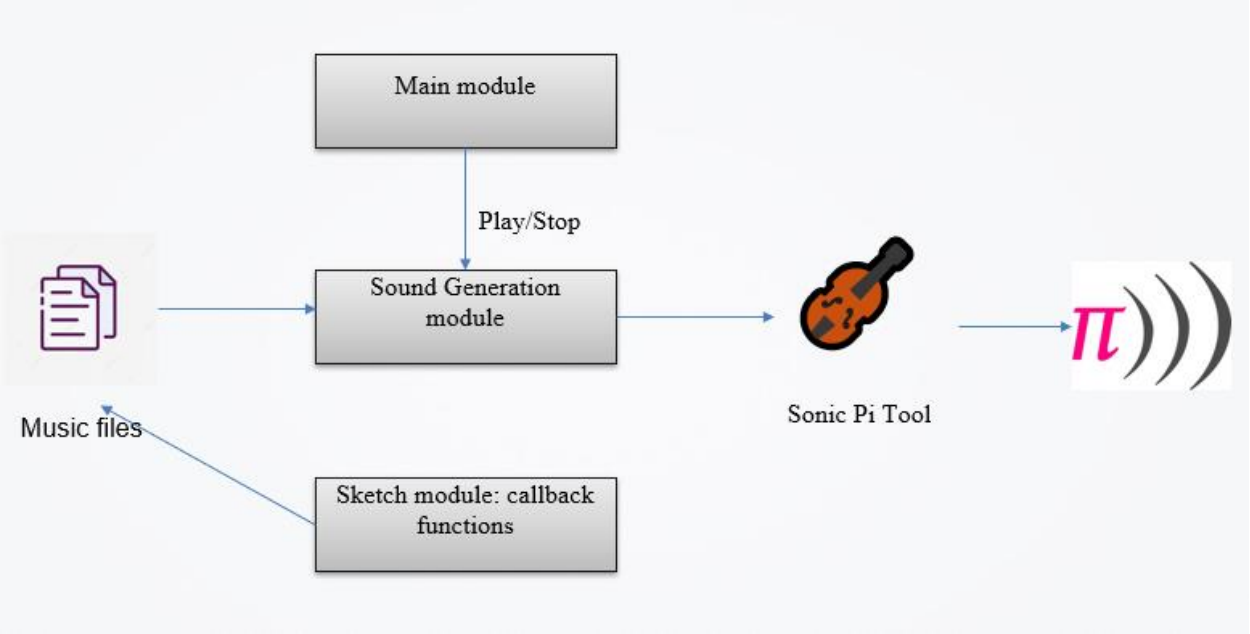


Figure 1. The three modules of the Emugel architecture.

The sketch module stores the definitions of the functions that handle the sounds. It defines all the data structures in the library that represent musical elements, such as events, durations, sound processing, synthesis instruments, modes and so on. It also specifies the callback functions to implement in the music-file modules. Therefore, when users write sketches as music-file modules, they need to follow the syntax of these callback functions described in the next section.

Lite coding syntax

This section illustrates the data structures of the sketch module, including the music elements as well as the callback functions featured in the sketch files.

Sketch. A sonic outcome is abstracted as a ‘sketch’ in our library. A sketch may comprise six functions: `beats_per_minute()`, `pitch_based_track()`, `chord_play()`, `sample_pattern_track()`, `rubato_pattern_track()`, `choose_play()`. These functions can be freely combined as independent tracks, supporting as many instances as the device memory permits.

Furthermore, the definition of these functions includes multiple data types: `note`, `beats`, `sound`, `envelope`, `envelope_attribute_name`, `envelope_attribute_value`, `name`, `effect`, `instrument`, `mode`, `delay`, `duration`, `sample`, `rest`, `hit`, `pattern`, `first`, `last`, `chord_type`.

Beats_per_minute(). The `beats_per_minute()` function is used to set the tempo of periodic sequences. It receives a non-negative integer, from 1 to 900, although normally we

expect users to use a value between 60 and 180 for it. If the value is above the threshold 900, Sonic Pi can throw a timing exception when playing the track, as the thread can get behind schedule at a high tempo. Especially for tracks that play consecutive events without breaks, scheduling may become flaky. Box 1 exemplifies its usage.

Box 1. Example of beats_per_minute() function.

```
beats_per_minute() ->
60.
```

Pitch_based_track(). The `pitch_based_track()` function features pitches and their durations. It is basically a list of tuples. Each tuple is a pitch track and includes `name, mode, duration, delay, instrument, sound, effect, envelope`. Box 2 is an example of its usage.

Box 2. An example of pitch_based_track() function.

```
pitch_based_track() ->
[
    {
    "melody",
     loop,
     5.075,
     10.00,
    "piano",
    [
    {b1, '_'},
    {b2, '__'},
    {e1, '__'},
    {e2, '____'},
    {b3, '__'},
    {60.5, '__'}
    ],
    no_effect,
    [{attack, 1.0}, {decay, 1.0}, {sustain, 1.0},
    {release, 1.0}, {amplitude, 0.1}]
    }
].
```

Name. The `name` type is defined as a string that labels the track.

Instrument. The `instrument` type is a string that defines the sound source of this track. Since we need to use Sonic Pi, all the instrument strings follow the 'synth' definition of Sonic Pi.

Mode. The `mode` type is related to the generative mode for the track. It can be either loop or non-loop. When the value is "loop", the track is iterated till the end of the track duration.

If the value is 1, the track is played only once. If it is an integer larger than 1, the track is iterated the number of times set by the user.

Delay. The `delay` type can be used to set the onset time. It can be either "no_delay" or a positive float number with at most three decimal places. If "no_delay" is given, the track will be played at the sketch onset. If a positive float number is given, the track playing will be delayed by the given duration.

Duration. The `duration` type is used to define the track duration for a single iteration. As with the `delay` type, when set to "no_duration", the track will include all the events. When a positive float number (three decimal places) is given, the events will be trimmed to fit with the given duration.

Sound. The `sound` type represents the events. A `sound` type is a list of tuples. Every tuple includes a `note` type and a `beats` type, which represent the pitch and duration of the event, for example, `{e1, '__'}` is the pitch class E on the first MIDI octave with a duration of two temporal units.

Note. The `Note` type is defined as a list of atom Enums. The value of the `Note` type may be a combination of a letter (pitch class) and a number (octave), a MIDI number (a float number to allow for microtones), or 'rest' to represent a pause. As can be seen from Box 3, each atom is a combination of letters and a number. The first letter represents a diatonic pitch, it can be 'a', 'b', 'c', 'd', 'e', 'f' or 'g'. The second letter is either an 's' or a 'b', 's' meaning sharp and the letter 'b' for flat. The number means the octave which ranges from 0 to 10, according to the MIDI standard. All atoms are in line with the representations required by Sonic Pi.

Box 3. Part of the Note type definition.

```
-type note() ::
c0 | cs0 | db0 | d0 | ds0 | eb0 | e0 | f0 | fs0 | gb0 | g0 | gs0
| ab0 | a0 | as0 | bb0 | b0 |
     c1 | cs1 | db1 | d1 | ds1 | eb1 | e1 | f1 | fs1 | gb1 | g1
| gs1 | ab1 | a1 | as1 | bb1 | b1 | …
     c10 | cs10 | db10 | d10 | ds10 | eb10 | e10 | f10 | fs10 |
gb10 | g10 | gs10 | ab10 | a10 |
as10 | bb10 | b10 | rest | float().
```

Beats. The `Beats` type is represented by the analogue-proportional notation: '_', '__', '___', '____', '_____', whose size correlates to added units of time. '_' is 0.25 or a quarter of beat, '__' is 0.5 or half a beat, '___' is 1 or a whole beat, '____' is 2 beats and '_____' is 4 beats.

Effects. The `effect` type is used to describe audio processing parameters. Its definition is a series of atoms. Currently, emugel supports reverb, echo, wobble, slicer, distortion, ping_pong, record and no_effect, no_effect indicating bypass.

Envelope. The `envelope` type represents an ADSR amplitude envelope, in line with the ADSR amplitude envelope in Sonic Pi. The `envelope` is either a list of tuples of envelope

Proceedings of UbiMus2023, II- Ubiquitous Music Symposium

attribute names and envelope attribute values, or an atom of `no_envelope`. `no_envelope` sets the default envelope value to the duration of one beat (one second with BPM 60). Given an `envelope` type, the duration time for one beat will be the sum of all the envelope attribute values except the `amplitude` value. The `amplitude` value sets the maximum amplitude of a `sound`, defined as a float. The envelope attribute name contains the atoms of `attack`, `decay`, `sustain`, `release`, `amplitude`, `attack level`, `decay level` and `sustain level`. Attack stands for the time of fading in. Release is the time to fade out. Sustain phrase is the time maintained at full amplitude between the attack and release phases. Release is the time to go from the sustain level to 0.

Chord_play(). The `chord_play()` function is a variation of `pitch_based_track()`. It is used to input pitched aggregates. In each track, users could define a chord to be played synchronously. This function is also a list of tuples. Each tuple is made of the type of `name`, `mode`, `delay`, `instrument`, `note`, `chord_type`, `beats` list, `effect` and `envelope` type.

The repeated types' syntax follows the `pitch_based_track()`. A `chord_type` is defined as a list of atoms that represents a collection of 3 or 4 pitches: `major`, `minor`, `augmented`, `diminished`, `major7`, `minor7`, `sus2`, `sus4`. While the chords `major7` and `minor7` feature 4 pitches, all other chords include just 3 pitches. This impacts the number of durations (`beats`). The `beats` set independent event durations for the elements of the chord. Thus, users need to give a `beats` list corresponding to the chord type: either 3 items for `major`, `minor`, `augmented`, `diminished`, `sus2`, `sus4` chords, or 4 items for `major7`, `minor7` (Box 4). The program will report an error if the user fails to do so .

Box 4. Example of chord_play() function.

```
chord_play() ->
[
   {
    "chord1",
    1,
     no_delay,
     "beep",
    e3, minor,
    ['___', '____', '_____'],
    no_effect,
    [{amplitude, 1.0}]
}
].
```

Sample_pattern_track(). The `sample_pattern_track()` function is used for recorded sounds in Sonic Pi, with a custom pattern to compose the track. It is a list of tuples. Each tuple represents a pattern track. Each track features `name` type, `mode` type, `duration` type, `delay` type, `sample` type, `pattern` type, `effect` type and `envelope` type. Below is an example of this track.

Box 5. An example of sample_pattern_track().

```
sample_pattern_track() ->
[
    {"sample1", 2, 6.500, 3.005, "drum_heavy_kick", [x, o, x, o,
x, x], no_effect, [{amplitude, 0.25}] },
    {"sample2", 1, 5.755, no_delay, "drum_snare_hard", [o, x, o,
x, o, o], no_effect, [{amplitude, 0.25}]}
].
```

Sample. There are two new types in this track: `sample` and `pattern`. The `sample` is defined as a string. It can be assigned a pre-recorded sample name in Sonic Pi, such as '`drum_heavy_kick`', '`drum_snare_hard`' and '`bass_dnb_f`. Users can also define their own samples by giving the full path to the sample location in the hard drive. The sample names in each tuple must be different. This is because each tuple will be transferred to a thread in Sonic Pi to play at the end. If two threads share the same name, Sonic Pi will only use the last thread, playing the last tuple.

Pattern. The second new type is `pattern` which defines a sequence of events. This is a list of `hit` and `rest` types. The `hit` type is defined as an atom '`x`', while the `rest` type is defined as an atom '`o`'. A hit represents an event while a rest represents not taking any action.

Rubato_pattern_track(). `rubato_pattern_track()` serves to input a rubato pattern. Besides using the samples and patterns, the rubato pattern track provides the transition from an initial duration to an end duration according to a sequence of 'xo' (action, no action). If the first event is short and the last one is long, then the result will be a rallentando, for instance, '_ __ ___ ____ _____ ______ _________'. In contrast, if the first event is long and the last one is short, the result is an accelerando, '_________ ________ _______ _____ ____ ___ __ _'.

`Rubato_pattern_track()` comprises a list of tuples as tracks, each track includes `name` type, `mode` type, `first` type, `last` type, `delay` type, `sample` type, `pattern` type, `effect` type and `envelope` type. The `first` sets the first beat duration and the `last` sets the last beat duration. They are represented as duration atoms: '_', '__', '___', '____', '_____'. Box 6 gives an example of usage.

Box 6. An example of the rubato_pattern_track().

```
rubato_pattern_track() -> [
  {"rubato1", 1, '_____', '_', 2.35, "drum_heavy_kick",
   [x, x, x, x, x, x, x, x, x, x, x, x, x], no_effect,
   [{amplitude, 1.0}]}
].
```

Choose play(). The function `choose_play()` takes advantage of the Sonic Pi random number generation. In this kind of track, two lists of pitches and durations are given. Then the program will randomly select an element from these lists separately, every time it executes the loop. As with the previous functions, this function is defined as a list of tuples. There is a `name` type, a `mode` type, a `delay` type, an `instrument` type, a list of `note` types, and a list

of `beats` types in each tuple. Differently from other tracks, users may give a list of `note` type instances along with a list of `beats` type instances of this track.

The following example includes 5 iterations. The possible results for note and beat combinations could be Iteration 1: {eb4, '_____'}, Iteration 2: {gb4, '_'}, Iteration 3: {eb4, '_____'}, Iteration 4: {65.89, '__'}, Iteration 5: {gb4, '_'}.

Box 7. Example of a choose_play() function.

```
choose_play() ->
[
   {
    "choose1",
    5,
    4.00,
     "piano",
     [eb4, 65.89, gb4],
     ['_', '_____', '_____', '__']
     }
].
```

Preliminary study

We have carried out experimental lite coding sessions with participants to identify caveats and usability issues related to the deployment of the emugel prototype.

Participants. Three participants were involved in the studies, 2 novices and 1 musician. They worked independently. The two novices both master object-oriented languages but are new to functional programming languages. They have no experience in music programming languages and obtain very little general music knowledge. The musician is skilled in music programming, has a strong technical background and is professional in functional languages.

Preparation. To allow the participants to grasp a basic understanding of emugel, we furnished an API documentation with detailed usage of the system. We also created a reference guide with audio examples for all the synthesis and processing tools in emugel. A template sketch file with code examples and default parameters and functions was provided as well. Thus, the participants only needed to modify the parameters when conducting the tasks.

Tasks. The study is designed to be a 45-minute session with 3 targeted tasks. Three track functions enable the tasks: `pitch_based_track()`, `sample_pattern_track()`, and `choose_play()`.

The participants are given 15 minutes to read the API to get familiar with the synthesis instruments and software functionality. The tasks start from the function `pitch_based_track()`. Participants are asked to change each parameter step by step. The subsequent task targets `sample_pattern_track()`, focused on pattern creation and exploration of the samples. Finally, during the `choose_play()` task the participants are

asked to try various pitch and duration combinations while exploring the application of randomness for musical purposes.

Findings. The sample pattern tracks with percussion furnish a good introductory activity for new users of the emugel system. All the participants managed to conduct the tasks within 10 minutes. The learning process of new tracks was gradually shortened after they became familiar with the basic syntax of emugel functions. Amateurs and musicians both managed to create various instances of this strategy within minutes by applying different sequences of 'xo' events to access the drum samples. Musicians showed a better grasp of pitch-based tracks than untrained subjects. Not surprisingly, they were more comfortable in using pitch-based compositional strategies and timbre definitions than the lay subjects. Emugel enables them to create melodies quickly by combining all types of tracks. The analogue-proportional notation was well accepted by all participants as a way to handle durations.

The syntax of Erlang tuples presented some problems for them. Given that we treat every track function as a list of tuples, when users wanted to include multiple tracks in the function `sample_pattern_track()`, they tended to forget to add a comma after each tuple and they also tended to forget about discarding the comma for the last tuple. This often resulted in execution malfunction, delayed progress in task completion and confusion of error cause. Since the emugel data structure is based on tuples and lists, this kind of error may emerge frequently in casual usage.

Another problem involves running and stopping the system commands. Due to the lack of technical background or detailed knowledge of the emugel architecture, some participants needed further guidance to issue prompt commands successfully. Several attempts were made before they understood the process of saving the sketches, copying and triggering the prompt commands, and listening to the sonic results. Besides, the ADSR envelope settings were rarely used by the participants when asked to compose their own music. The amateurs would only modify a single amplitude attribute when dealing with dynamics.

Discussion

This paper presents key findings of an initial deployment of the Erland Music Generation Library 1.0 (emugel). The emugel prototype provides support for programming audio synthesis and processing techniques by untrained subjects. We label this form of music programming as *lite coding*. Our findings confirm the feasibility of the proposal and highlight some caveats that need to be considered in future iterations of the emugel design cycle.

A prompt-based interface along with a preset template sketch file provided a concise and easy-to-start material for the participants. All subjects were able to implement simple sketches with good sonic results within the strict temporal constraints of the proposed activities. A tendency to shorten the temporal investments was observed throughout the study, pointing at the ability of the participants to transfer the acquired knowledge to different tasks. Furthermore, among the proposed strategies *xo notation* was the most effective means to achieve sonic results.

We observed two types of drawbacks: (a) Errors that precluded the successful completion of the task, and (b) Errors that reduced the range of creative outcomes but did not stop the subjects from obtaining sonic results. *Type a* errors included Erlang-specific requirements, such as the use of commas and handling the prompt-based interface. The error

messages returned in the prompt were not easily understood by novice users. Also, some subjects had difficulty interpreting the written information featured in the documentation. Our next design iterations will approach these issues by producing simple audiovisual instructions to facilitate training. Hopefully, an instructional strategy based on audiovisual examples may be more effective.

Type b errors highlight the areas where our approach to lite coding needs refinement. Both the analogue-proportional notations employed for durations and the xo notations employed for sequences of percussive events were quickly and effectively incorporated within the subjects' creative cycle. But the envelope values for dynamics and the letters and numbers proposed for sequences of pitches were not readily accessible to non-musicians. We will need to design lite-coding approaches that provide simpler representations of these parameters without impacting the pliability of emugel design. In fact, aesthetic pliability is one of the dimensions we did not have a chance to explore, despite being among the central requirements of ubimus ecosystems. Our forthcoming studies will address this gap.

References

Aaron, S., Blackwell, A. F., & Burnard, P. (2016). The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology & Education*, **9**(1), 75-94.

Aliel, L. S., Keller, D., & Costa, R. L. M. (2018). The Maxwell Demon: a proposal for modeling in ecological synthesis in art practices. *Música Hodie*, **18**(1), 103-116.

Armstrong, J. (2010). Erlang. *Communications of the ACM*, **53**(9), 68-75. New York: ACM.

Armstrong, J. (1996). Erlang—a Survey of the Language and its Industrial Applications. In *Proceedings of International Conference on Applications of Declarative Programming and Knowledge Management* **96** (pp. 16-18). Tokyo: INAP.

Bessa, W. R. B., Keller, D., Silva, J. B. F., & Costa, D. F. (2020). A metáfora da esfera sonora desde a perspectiva WYDIWYHE. *Journal of Digital Media & Interaction*, **3**(5), 60-88.

Keller, D., Messina, M., & Oliveira, F. Z. (2020). Second wave ubiquitous music. *Journal of Digital Media & Interaction*, **3**(5), 5-20.

Kramann, G. (2020). Composing by laypeople: A broader perspective provided by arithmetic operation grammar. *Computer Music Journal*, **44**(1), 17-34.

Mathews, M. V., Miller, J. E., Moore, F. R., Pierce, J. R., & Risset, J. C. (1969). *The technology of computer music*. Cambridge, MA: MIT Press.

McCartney, J. (2002). Rethinking the computer music language: SuperCollider. *Computer Music Journal*, **26**, 61–68.

Messina, M., Keller, D., Aliel, L., Gomez, C., Célio Filho, M., & Simurra, I. (2022). The Internet of Musical Stuff (IoMuSt): ubimus perspectives on artificial scarcity, virtual

communities and (de)objectification. *INSAM Journal of Contemporary Music, Art and Technology*, **9**, 26-50.

Messina, M., Svidzinski, J., de Menezes Bezerra, D., & Ferreira, D. (2019). Live Patching and Remote Interaction: A Practice-Based, Intercontinental Approach to Kiwi. *CMMR 2019 and Workshop on Ubiquitous Music (UbiMus 2019)* (pp. 696-703). Marseille: Ubiquitous Music Group.

Miletto, E. M., Pimenta, M. S., Bouchet, F., Sansonnet, J. P., & Keller, D. (2011). Principles for music creation by novices in networked music environments. *Journal of New Music Research*, **40**(3), 205-216.

Stolfi, A. S., Milo, A., & Barthet, M. (2019). Playsound. space: Improvising in the browser with semantic sound objects. *Journal of New Music Research*, **48**(4), 366-384.

Python Software Foundation (2014). Welcome to Python.org. (n.d.). Beaverton, OR: Python Software Foundation. Retrieved August 17, 2022, from https://www.python.org/

Dan, G., Björn, G., Nendo & Mirai (2023). Wings 3D. Retrieved August 17, 2022, from http://www.wings3d.com/

Wright, M., Freed, A., & Momeni, A. (2017). 2003: Open Sound Control: State of the Art 2003. *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression* (pp. 125-145). Berlin: Springer.

Paper Session

Ubimus Ecologies

Aesthetic emotions in a mixed reality multisensory experience with food crossmodally matched to music and visuals

Bruno Mesz[1], Jean-Christophe Sakdavong[2], Sami Silén[3], Anu Hopia[3]

[1]Universidad Nacional de Tres de Febrero (UNTREF). Instituto de Investigaciyn en Arte y Cultura (IIAC). Juncal 1319, CABA, Argentina. bmesz@untref.edu.ar
[2]Cognition, Languages, Language and Ergonomics Laboratory (CNRS UMR5263, Universitй de Toulouse), 5 Allйes Antonio Machado, 31058 Toulouse Cedex 9,France
[3]Functional Foods Forum, University of Turku, Finland

Abstract. We examine the emergence of aesthetic emotions in an exploratory study on a mixed reality (MR) gastrosonic experience. Participants in an art-science event completed the AESTHEMOS questionnaire on aesthetic emotions and the Multimodal Presence Scale, reporting a wide spectrum of prototypical, pleasant and epistemic emotions. Food enjoyment was correlated with the prototypical aesthetic emotions of fascination and enchantment and feelings of self-presence were correlated with the prototypical emotions of being moved/ in awe. Crossmodally matched food was highly congruent with music and visuals. Our findings suggest the relevance of studying environmental aesthetics in the context of these novel eating situations.

Keywords: gastrosonics, food experience, mixed reality, aesthetics, emotion

1. Introduction

1.1. Gastrosonics in Ubimus

Throughout history, food has been associated with music. However, only recently have researchers started to consider the impact of what we hear on the experience of eating and drinking. Altering people's taste experiences by the use of sound has been called “sonic seasoning” (Reinoso-Carvalho et al., 2020; Galmarini et al., 2021). In particular, crossmodal correspondences between taste and sound have been used to modify the experience of many different

food and drink products by changing the music or soundscape that people listen to. Crossmodal correspondences refer to the associations that many of us appear to make between seemingly unrelated attributes, features or dimensions in two or more different senses (Spence, 2011).

In the context of Ubimus there has been a recent upsurge of interest in investigations on the interactions between technology, music and gastronomy, a field for which Ubimus researchers have proposed the name *gastrosonics.* (Note that this term is intended to have a different meaning than "sonic seasoning" since the latter refers specifically to perceptual or cognitive aspects of eating or drinking in the presence of sound). This has led to several publications and a special dossier on "Ubimus, Gastrosonics and Well-Being" in the Vyrtex Journal (Keller et al, 2021). For instance, Keller et al. (2022) report a preliminary crossmodal study using drinks prepared at home as triggers for sonic choices. The basic sonic materials included sounds that were previously classified as crossmodally corresponding to various tastes, or as neutral. The task was to create a sonic mix with segments crossmodally congruent to each drink; this was done in a collaborative fashion among small groups of participants. Mesz et al. (2023) report a gastrosonic commensality experience for three guests, called *Quorum Sensing*. A soup was served in a glass tableware, and the actions of each diner with the spoon produced vibrations applied to the body of the others, varying in intensity depending on the rate of movement of the spoon.Still another example is the system "Taste this score" of Rosales (2023), where the composer elaborates a videoscore including visual textures of food, proposing the realization of sonic dinners by making scores from the visual textures of each dish and interpreting them to enrich the eating experience.

In the present study, we examine the aesthetics of gastrosonic experiences in immersive mixed reality environments, which are likely to grow in popularity in the near future, also allowing researchers to conduct more ecological studies in the laboratory. In the next section we characterize these environments and provide some background on their use in the context of gastrosonics.

1.2. Virtual reality (VR) and mixed reality (MR) gastrosonic experiences

A definition of virtual reality (VR) can be "an integration of several elements, including computers, worlds and environments, interactivity, immersion, and users, who are usually referred to as participants in a virtual reality experience." (Muhanna, 2015).

Augmented reality (AR) is a general term used to characterize experiences where the reality is augmented by supplementary information: it can be seen using a headset inside a museum or through a projected display on the shield of a modern car. Usually, information brought by AR is simple, e.g. the speed of the car. Mixed Reality (MR) is a more modern term meaning that a part of the real world (in addition to the user) is mixed with the virtual world. The virtuality continuum from Milgram and coworkers (1994) shows how virtuality and reality can be interlaced and Figure 1 is a revisited view of this virtuality continuum using food. This figure shows how we can include real foods (with their specific sensory properties: texture, smell and taste) in a virtual environment.

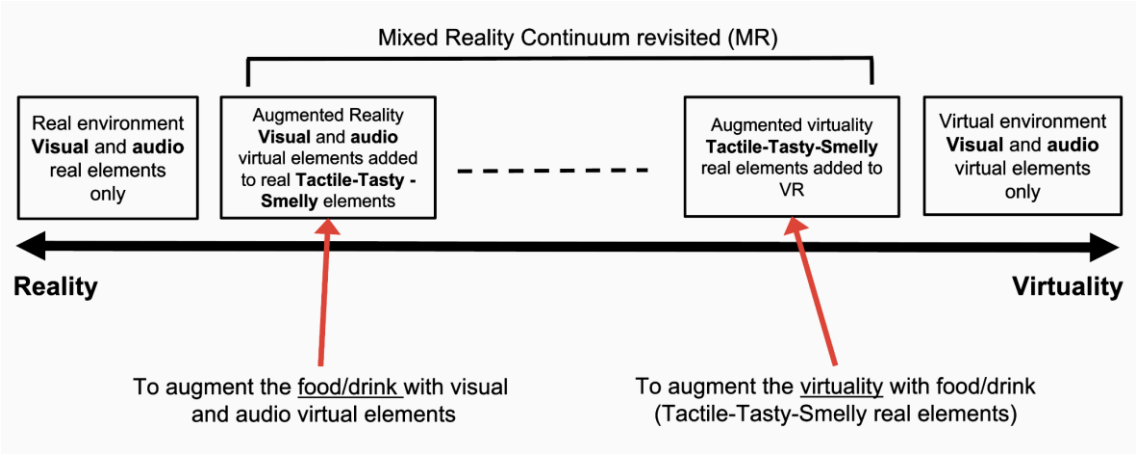


Figure 1. Virtuality continuum revisited.

Virtual reality (VR) started to be used in food related research a few years ago, with the first published studies dating back to the late 2010s, see e.g. Wang and coworkers (2021). The first application of VR was to replace sensory booths in sensory and consumer studies. The second step when using VR or MR for food experiences is to use them in order to change the perception of the food or its enjoyment. Two ways of doing this are altering the food aspect or changing the environment, for example, by changing the color of the food (Ammann et al, 2020; Wang et al, 2020), or altering the color, texture and shape of the environment (Chen et al., 2020).

Regarding gastrosonics, a recent study (Nivedhan et al, 2020) demonstrated that cold brewed coffee tastes sweeter when consumed in a virtual environment with a pleasant color and music, compared to one with an unpleasant color and music.

1.2.1. Presence and its measure

An important feature potentially affecting aesthetics and emotion in immersive experiences such as the one we describe here is presence. Presence is the feeling of being inside the displayed virtual environment (Slater and Wilbur, 1997; Felton and Jackson, 2022; Wilkinson et al, 2021). This concept is widely used in studies where a user interacts with an immersive virtual environment using a virtual reality headset. Studies like Riva and coworkers (2007) and Chirico and coworkers (2017) have shown that presence in VR or 360° videos emerged as an amplifier for emotional responses.

Presence is also known as an important factor enhancing enjoyment in many activities such as video games. We hypothesized that it is likely that this also holds true for VR/MR food experiences. The concept of presence is multidimensional and researchers like Lee (2004) or Makransky et al. (2017) use the concepts of physical presence (e.g. the sense of physical realism, the absence of attention toward the outside world, the non-awareness of physical mediation), social presence (e.g. the feeling of coexisting with other beings) and self-presence (e.g. the sensation of extension and prolongation of the body of the user in the avatar displayed in the virtual environment). Makransky et al. (2017) have developed the Multimodal Presence Scale in order to measure the three dimensions of presence according to Lee: physical presence, social presence and self-presence. This three-dimensional scale is obtained using a questionnaire of 15 questions (5 questions per dimension).

1.3. Aesthetic emotions in MR gastrosonic experiences

Aesthetics is an important area of study in ubiquitous music, especially in the context of everyday interactions with technology (Keller et al, 2015 a, 2015b); However, aesthetic emotions, i.e. the emotions that can arise when a person perceives and evaluates a stimulus for its aesthetic appeal or virtue (Menninghaus et al., 2019) have hardly been investigated in connection with gastrosonics, and more generally, the literature on aesthetics of food experiences is scarce. For instance, only a few studies consider the aesthetic experience of eating in restaurants and its effect on food perception and enjoyment, going beyond the sensory and atmospheric aspects to consider aspects such as beauty (Marković et al, 2021). In the case of the present study, the MR environments were designed as a gastrosonic art installation for a public art and science event, and our focus of interest were precisely aspects such as the emergence of aesthetic emotions and food enjoyment.
Schindler and coworkers (2017) developed a questionnaire for aesthetic emotions (AESTHEMOS) consisting of short expressions, applicable across different domains. Examples of items in the scale are: “I found it beautiful,” “Delighted me,” “Bored me” and “Challenged me intellectually”. In another study (Beermann et al., 2021) the authors identify 15 terms for emotions from clusters derived from a pool of 75 aesthetic emotion terms that were used to develop the AESTHEMOS questionnaire: Moved / In awe, Longing / Melancholic, Invigorated / Interested, Confused / Worried, Merry / Attracted, Fascinated / Enchanted, Intellectually Stimulated, Relaxed, Surprised, Sad, Pleased/feeling harmony, Displeased / Repelled, Delighted / Feeling beauty, Bored, Angry. A different classification is given by Schindler et al. (2017), who group the AESTHEMOS questionnaire in four broad emotion subclasses: prototypical (feeling of beauty/liking, fascination, being moved, and awe), pleasant (joy, humor, vitality, energy, and relaxation), epistemic (surprise, interest, intellectual challenge, and insight) and negative (feeling of ugliness, boredom, confusion, anger, uneasiness, and sadness).

The AESTHEMOS questionnaire has been used in research on emotions in the visual arts (Hosoya, 2021), film (Frцber and Thomaschke, 2020), music, architecture and nature (Egermann and Reuben, 2020). A recent study (Diessner et al., 2021) discusses the relevance of aesthetic emotions for gustatory and olfactory experiences, finding the AESTHEMOS well suited to determine the aesthetic emotions elicited by scents and tastes.

Based on the literature mentioned above, which demonstrates the wide applicability of this questionnaire, we decided to use it in the context of our experiment in mixed reality involving food, visuals and sound art.

1.4. Aims and main findings of the study

It was the aim of this exploratory study to examine the impact of environmental aesthetics on aesthetic emotions, feelings of presence, perceived congruence between environment and food, and food enjoyment in a mixed

reality (MR) gastrosonic experience. The experience was divided in two clearly differentiated parts (A and B). All participants were presented with both parts in the same order (first part A, then part B). In part A food was more standard and familiar (Gouda cheese and Malbec wine), the aspect of the food was realistically represented in VR, and participants remained seated when eating and drinking. The music was designed to be crossmodally congruent with the food, and visuals were inspired by bacterial processes in wine and cheese. In part B it was the food instead that was elaborated to be crossmodally correspondent with the visuals, music and simulated temperature of the environments; its appearance was disguised as small "planets" which moved in space and participants were induced to stand up and turn around to pick the food and contemplate the environment. The main difference between parts A and B was the familiarity vs. unfamiliarity of the food aspect and behavior. We can summarize our main findings as follows:

- A wide spectrum of aesthetic emotions were elicited in the MR food experience.
  - These aesthetic emotions mostly had positive valence and the subclass of negative aesthetic emotions was scarcely represented.
  - Most participants experienced feelings of beauty
- Food enjoyment was correlated with emotions of fascination and enchantment in Part A.
- In Part B, perceived congruence of crossmodally designed food with music and visuals was high, according to our expectations.
- In Part B, unusual looking food that moved around in space was highly enjoyed.

2. Materials and Methods

2.1. Participants

Twelve volunteers (3 males, 9 females, aged from 26 to 66 with a mean age of 41.9, and an SD of 13.6) participated in the study. Due to the assembly restrictions during the Covid pandemic they were participants by invitation to the art and science event "Do science, art and technology spice up future food experiences?" and they represented the employees of the University of Turku.

2.2. Stimuli

Participants were supplied with an Oculus/Meta Quest 2 VR headset - see web reference "Food Experiment Turku December 2021". During the whole experiment, the participant was seated in front of a real table with a dish with cheese cubes, a 3D printed glass of wine and a shelf with balls of food (cf. Figure 2a). The cheese samples were of a mild gouda-type cheese (Brandy-рдhкiпд by Juustoportti Ltd., Jalasjдrvi, Finland) and the wine organic Alamo Malbec 2020 from the Mendoza region, Argentina.

In Part A, the virtual reality showed the same table, dish, cheese and glass of wine as in the real world. When the participant was not touching the food, the background was black. When the participant touched the food, 360° videos were displayed as a background. This visual background was designed by the Dutch-Argentine artist Rob Verf (Rob Verf 2021, 2021B) . He produced two 360° videos inspired by the microbial fermentation landscapes of cheese

and wine. These videos were triggered by the action of taking hold of a piece of cheese or a glass of wine, respectively, and continued for 20 seconds after this action (in the case of cheese) or after the glass was placed back on the table (for the wine). The appearance of the food and wine were designed by Jean-Christophe Sakdavong with the help of the AD2RV non-profit association using Unity 3D and Liquid Volume pro 2 assets. The location of the 3D printed glass of wine and the cubes of cheese was precisely mapped in order to have a perfect synchronization between virtual reality and the real world. The glass was designed in order to lower finger occlusion while the glass was being held.

In Part B, "Planets of Fire & Ice", the visual background represented imaginary cold and hot outer space environments. The environments and appearance of the foods were designed by Jean-Christophe Sakdavong, with AD2RV. Realistic effects pack 3, AOE magic spells, Aura and ground effects were used. The food for Part B were cold chocolate balls and warm cheese balls described in the supplementary material (S1). In the VR/MR installation the food balls ("the planets") were placed on a specially designed shelf with 3D printed spoons, which were precisely mapped in order to have perfect synchronization between virtual reality and the real world.
A video showing the environments and interaction can be found at Food Experiment TURKU December 2021

2.2.1 Crossmodally matched music

During Part A, when the 360 videos were not active, soft conversation noises were played. When the cheese-related video was active in the background, the music combined electro-acoustical processed sounds of kombucha fermentation intended to be semantically related to the microbial world (the sounds had been previously played and its origin explained to the audience during the event) and to create a feeling of weirdness, mixed with soft, legato consonant chords related to the dominant emotion and sensory attributes associated to Gouda cheese according to Schoutetten et al. (2015): pleasantness, softness and creaminess. When designing these chords, taste pleasantness was associated with psycho-acoustic pleasantness (consonance), texture softness with soft sound dynamics, and creaminess with legato (Reinoso-Carvalho et al., 2017). In contrast, when the wine was taken and the wine-related video was playing, the music consisted of a melancholic sound texture to match an aged red wine (Spence and Wang, 2015). In Part B, when the "cold deep space" environment was present, the music was composed of a high-pitched strong wind sound and high-pitched dissonant chords, reflecting cross-modal correspondences of coldness with high pitch (Spence, 2020). Moreover, a sound texture of breaking ice was synchronized with the act of taking the cold eucalyptus-menthol-chocolate balls with a rough surface and crunchy texture. This music, together with the visuals, transitioned to a "hot deep space" environment, populated by a low-pitched bubbling sound, a soft low-pitched wind and cross-modal congruent medium-pitched chords (Wang and Spence, 2017). A sizzling sound in a frying pan was synchronized with the taking of the warm jalapeco cheese balls (recipes can be found in Supplementary Material S1).

The music was designed by Bruno Mesz and Sami Silйn; sounds of fermentation were recorded with two large diaphragm microphones.

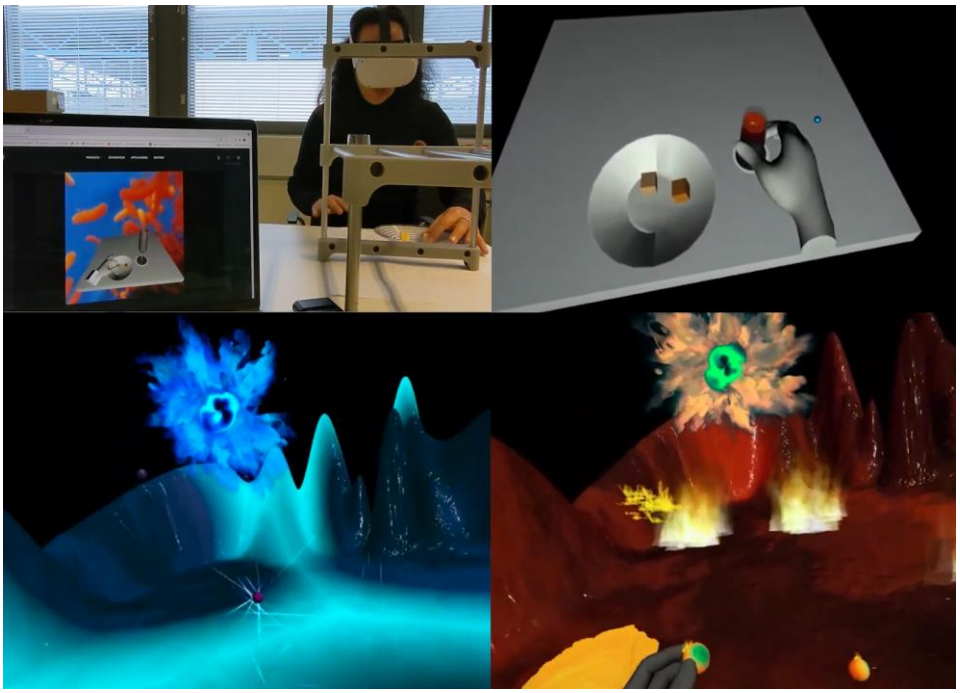


Figure 2. From left to right, top to bottom: (a) Setup, (b) Part A, (c) Part B cold planet environment, (d) Part B hot planet environment.

2.3 Experimental procedure

The experiment was carried out during an art-science event at the Aistikattila multisensory laboratory, located at Flavoria in Turku University in Finland (see web reference "Webinar in Turku"). The Flavoria study protocol was reviewed and ethically approved by the Ethics Committee for Human Sciences at the University of Turku, Humanities and Social Sciences Division (37/2021). The study followed the European Union's General Data Protection Regulation (GDPR). For Part A the questionnaire comprised:

(i) Two food enjoyment questions where we asked participants to rate the affirmations "I enjoyed drinking the wine (eating the cheese)" with numeric 5-point scales. For these scales, the lowest value, 1, was interpreted as 'completely disagree', and the maximum, 5, as 'strongly agree'.
(ii) A list of the terms from the AESTHEMOS questionnaire, asking participants to check all the terms corresponding to their experienced emotions, that is, a Check-All-That-Applies (CATA) task. Responses were stored in dichotomous variables, one for each term, having the value of 1 if checked, 0 otherwise.

For Part B we formulated the same questions (i) on food enjoyment, related here to the chocolate and cheese balls (denoted in the questionnaire by "food in the cold planet" and "food in the hot planet" respectively), and (ii) on aesthetic emotions, and moreover added the following items:
(iii) The MPS scales with 10 questions about physical presence and self-presence (as we omitted the 5 questions regarding the "social presence"

dimension, which were not relevant here because there was no social interaction). The answers were rated with a 5-points Likert scale from -2 to +2 and used to compute two variables, MPS_PHYS and MPS_SELF (see Section 3.2).
(iv) Four congruence rating questions: congruence between the visual and the music environments and the chocolate and cheese balls in 5-point scales from 1, “completely disagree” to 5, “strongly agree" that they are congruent. From the responses we computed two variables, averaging the visual and music ratings for each food, and a third variable CONGRUENCE FOOD-ENVIRONMENT B as their mean.
(v) Four questions to evaluate the subjective temperature of the VR environment before and during eating the chocolate and cheese balls, using 5-point scales from 1, “very cold” to 5, “very hot”.

The experiment took, in total, about 15 minutes for each participant.

3. Results and Discussion

Data was analyzed with the free and open statistical software Jamovi 2.2.5. and with MATLAB R2017b.

3.1 Aesthetic emotions

We evaluated the aesthetic emotion profiles of Part A and B by the proportion of participants that ticked, for a given emotion cluster, at least one AESTHEMOS term belonging to that cluster (Figure 3).

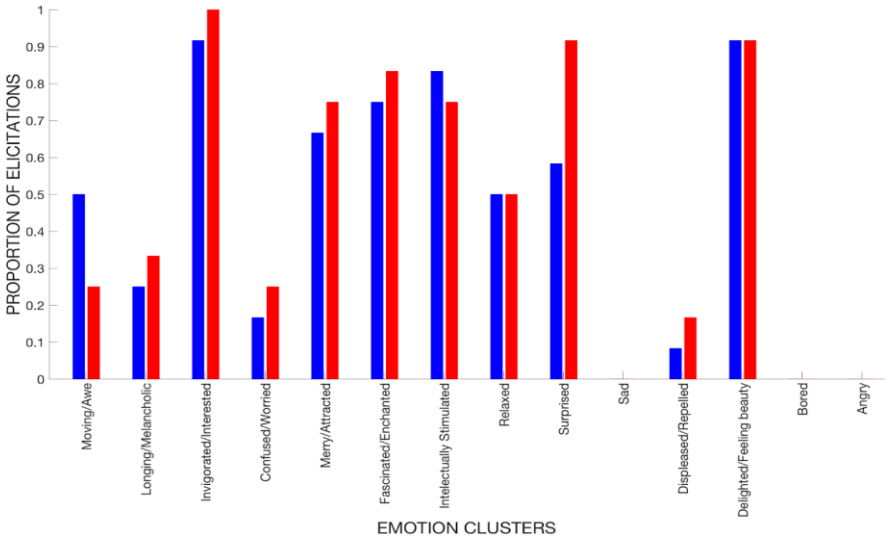


Figure 3. The aesthetic emotion profiles of the participants based on proportion for Part A (blue bars) and B (red bars) (1 representing that 100 % of the participants ticked an AESTHEMOS term from that cluster).

Only small differences in individual emotion clusters can be observed between parts A and B of the experience, although the atmospheres and food

in both parts were quite different. Of the four broad subclasses of emotions considered in Schindler et al. (2017), the prototypical, pleasant and epistemic emotions were predominant in both parts of the experiment. The subclass of (so-called) negative aesthetic emotions (Confused/Worried, Displeased/Repelled, Sad, Bored and Angry) was scarcely represented, with no emotions selected in the last three negative categories. Somewhat similarly, in the study by Diessner and coworkers (2021), less proportion of negative emotions of the AESTHEMOS were elicited by perfume and food items in comparison with a control condition without chemical stimuli. In contrast to that study, however, epistemic emotions such as intellectual stimulation and surprise were also relevant in the present experiment, possibly reflecting the relative novelty of the experience and the fact that all the participants had just heard presentations on VR/MR dining and some of them were researchers in fields related to food science. Evaluating the CATA responses using the Cochran's Q test for the 11 emotion clusters having terms effectively selected within each part of the experiment (that is, excluding the Sad, Bored and Angry clusters), we found that the responses were significantly different within part A and B ($p<.001$).

Between parts A and B, pairwise McNemar tests showed that only Surprise A and Surprise B had a significant difference ($p = 0.046$).

3.2 Presence

We evaluated presence using the multimodal presence scale regarding physical presence and self-presence.

Each item was rated from -2 to +2, and the two dimensions MPS_PHYS and MPS_SELF were computed as the sum of their respective 5 items; thus, they took values between -10 and +10.

The descriptive statistics show that both dimensions were positive (MPS_PHYS: M= 2.33, SD= 3.75; MPS_SELF; M= 2.75, SD= 2.93), indicating that the participants had a feeling of presence.

3.3 Food enjoyment, Congruence, Temperature

The enjoyment of the cheese and wine in Part A was high (cheese: M=4.17, SD=1.337; wine: M= 4.67, SD=0.49), and not significantly different between the food types ($t(11) =-1.15$, $p =0.27$). In Part B also the enjoyment of the food in cold planet, chocolate balls (M=4.33, SD=0.98) and food in hot planet, cheese balls (: M=4.58, SD=0.79) was high and similar ($t(11) =-0.76$, $p =0.46$).

Average food enjoyment was similar between parts A and B ($t(11)=-0.14$, $p = 0.88$).

Congruence between the cold planet food (chocolate balls) and the VR environment was also high, both with respect to the visuals (M = 4.08, SD = 0.79) and music (M = 4, SD= 0.6). Similarly, congruence between the hot planet food (cheese balls) and the VR environment was consistently high for the visuals (M = 4.33, SD =0.98) and music (M= 4.17, SD = 1.03).

As mentioned in Section 2.3, in Part B we computed two variables, TEMP_COLD and TEMP_HOT, as the mean of the subjective temperatures of the "cold" and "hot" environments before and during eating. TEMP_COLD (M =2.75, SD =0.62) compared to TEMP_HOT (M=3.75, SD= 0.5) was

significantly colder ($t(11)=-3.94$, $p = 0.002$), which was expected and consistent with the high congruence experienced between the food and the cold and hot planet environments.

3.4 Correlation structure between the variables in Part A

We considered the food enjoyment variable ENJOYMENT A and 11 aesthetic emotion variables, (MOVED/AWE A, LONGING/ MELANCHOLIC A, etc.), one for each of the 11 emotion clusters having at least one tick. These variables were defined as the proportion of checked AESTHEMOS terms belonging to the respective cluster, relative to the total number of terms in that cluster.

We found a positive correlation between food enjoyment and FASCINATED / ENCHANTED emotions ($r = 0.626$, $p = 0.029$), and a negative correlation between food enjoyment and CONFUSED/ WORRIED emotions ($r = -0.602$, $p = 0.039$) which became non-significant after Bonferroni correction; more importantly, the effect sizes are large (Nakagawa, 2004).

We hypothesize that these correlations may show emotion transference (Spence and Gallace, 2011; Spence, 2021), which refers to the carrying over of positive feelings about one stimulus (in this case, the VR environment) to positive feelings about a different one experienced at the same time (cheese and wine), this being reflected in the pleasure of eating and drinking (Nivedhan et al., 2020). By emotion transference, the fascination of an unusual virtual environment -an emotion reported by 75 % of participants- may have enhanced the delight of the food experience.

3.5 Correlation structure between the variables in Part B

We found a large effect size correlation ($r = 0.738$, $p=0.006$) between the self-presence (variable MPS_SELF) and the "MOVED / AWE" cluster (see S3 for the Pearson correlation matrix of the response variables in Part B), which is consistent with a result shown in a study by Chirico and coworkers (2018): the feeling of presence and the "awe" emotion are bound. In our context, this can be interpreted as the fact that increased presence for place-based stimuli engender greater awe by increasing the perceived vastness and need for accommodation of the stimulus (Kahn and Cargile, 2021).

However, the congruence between food and environment (visual or auditory congruence or the mean congruence between the two modalities) was not significantly correlated with average food enjoyment, nor with enjoyment of cheese or chocolate specifically. In fact, there was no significant correlation between food enjoyment or food-environment congruence and any other variable, including all aesthetic emotions.

We also found a large effect size negative correlation between the perceived temperature of the chocolate balls and the cluster RELAXED ($r = -0.718$, $p = 0.009$), which is difficult to interpret, since the emotional response was referred to the whole of Part B of the experience.

Finally, temperature of the “cold” environment before eating the chocolate balls was correlated with temperature while eating them: $r = 0.618$, $p=0.028$, which suggests that the atmosphere anticipated the temperature during eating (at least in retrospective evaluation); with cheese this did not happen ($r = 0.34$, $p = 0.27$).

In summary, we did not find evidence consistent with an impact of the VR "outer space" environment on the emotionality or enjoyment of food experience. However, food enjoyment was high even if its non-edible aspect and simulated movement were unfamiliar, which may have been enhanced by the presence of positive arousing emotions (Motoki et al, 2022).

3.6. The case of Beauty

The applicability of the prototypical aesthetic emotion of beauty to food experiences has been an object of interest in philosophy and aesthetics at least since Aquinas.

Looking at responses to the first item of AESTHEMOS ("I found it beautiful"), we found that 10 out of 12 people (83%) found Part A beautiful, and 8 out of 12 people (66%) found Part B beautiful. As the response was related to the whole mixed reality experience, we cannot assess how this evaluation could be attributed more specifically to the food and/or VR environment components. We did not find a significant correlation between food enjoyment and the proportion of emotions selected from DELIGHTED/FEELING BEAUTY cluster (see Section 3.4); however (see Section 3.4) we found in Part A of the experience large effect size correlations between food enjoyment and a different group of prototypical aesthetic emotions, the FASCINATED/ENCHANTED cluster. A future study to resolve the emotional impact of these two factors (food and VR context) is warranted.

4. Conclusions

The use of the AESTHEMOS questionnaire in this study has shown that our mixed reality food experience was able to elicit a wide spectrum of aesthetic emotions, mostly non-negative ones. In particular, feelings of beauty, that are an age-honored object of debate in the context of food appreciation, were experienced by most of the participants. Epistemic emotions such as intellectual interest and surprise were also prominent. Moreover, we found a positive correlation between food enjoyment and emotions of fascination and enchantment in Part A. The feeling of presence in the virtual environment was also correlated with the emotions of being moved/in awe in Part B. Food enjoyment was high in Part B of the experience and not significantly different from food enjoyment in Part A , although food appearance in Part B was unusual. This result was perhaps due to the playful interaction and positive arousing aesthetic emotions. In view of this, it could be interesting to explore the use of MR to, for example, alter the aspect of certain foods to make them more acceptable, overcome food neophobia and helping people to accept special diets by changing food aspect and environment.

Funding: This research was funded by Kone Foundation, grant number 5 201905865 2.

Acknowledgements: We would like to thank the AD2RV non-profit association for the Unity development (https://www.ad2rv.fr/). We acknowledge the Kone Foundation for funding a significant part of the study. Nanna Rintala is acknowledged for the designing of the test foods, and together with Jaakko Rдkkцlдinen for preparing the foods.

References

Ammann, J.; Stucki, M.; Siegrist, M. True colours: advantages and challenges of virtual reality in a sensory science experiment on the influence of colour on flavour identification. *Food Qual. Prefer.* **2020**, *86*, 103998. https://doi.org/10.1016/j.foodqual.2020.103998.

Beermann, U.; Hosoya, G.; Schindler, I.; Scherer, K. R.; Eid, M.; Wagner, V.; Menninghaus, W. Dimensions and Clusters of Aesthetic Emotions: A Semantic Profile Analysis. *Front. Psychol.* **2021**, *12*, 1949. https://doi.org/10.3389/fpsyg.2021.667173.

Bessa, W. R. B., Keller, D., Farias, F. M., Ferreira, E., Pinheiro da Silva, F. & Pereira, V. S. (2015) SoundSphere v.1.0: Documentação e análise dos primeiros testes. In F. Z.

Oliveira, D. Keller, J. T. de Souza Mendes da Silva & G. F. Benetti (eds.), Anais do Simpósio Internacional de Música na Amazônia (SIMA 2015). Porto Velho, RO: UNIR., 2015. https://soundsphere.com.br/beta

Carvalho, F. R., Wang, Q. J., Van Ee, R., Persoone, D., & Spence, C. (2017). "Smooth operator": Music modulates the perceived creaminess, sweetness, and bitterness of chocolate. *Appetite*, 108, 383-390.

Chen, Y.; Huang, A. X.; Faber, I.; Makransky, G.; Perez-Cueto, F. J. A. Assessing the influence of visual-taste congruency on perceived sweetness and product liking in immersive VR. *Foods* **2020**, 9, 465. https://doi.org/10.3390/foods9040465.

Chirico, A.; Cipresso, P.; Yaden, D.B.; Biassoni, F.; Riva, G.; Gaggioli, A. Effectiveness of Immersive Videos in Inducing Awe: An Experimental Study. *Sci Rep*. **2017**, 7, 1218. https://doi.org/10.1038/s41598-017-01242-0.

Chirico, A.; Ferrise, F.; Cordella, L.; Gaggioli, A. Designing Awe in Virtual Reality: An Experimental Study. *Front. Psychol.* **2018**, 8, https://doi.org/10.3389/fpsyg.2017.02351.
Costalonga, L. (2015b). Interaction aesthetics and ubiquitous music. *Creativity in the digital age*, 91-105.

Crisinel AS, Spence C. 2012. The impact of pleasantness ratings on crossmodal associations between food samples and musical notes. *Food Quality and Preference*, 24, 136-140.

Diessner, R.; Genthôs, R.; Arthur, K.;Adkins, B.; Pohling, R. Olfactory and gustatory beauty: Aesthetic emotions and trait appreciation of beauty. *Psychol. Aesthet. Creat. Arts.* **2021**, *15,* 38-50. https://doi.org/10.1037/aca0000262.

Egermann, H.; Reuben, F. "Beauty Is How You Feel Inside": Aesthetic Judgments Are Related to Emotional Responses to Contemporary *Music. Front. Psychol.* **2020**, *11*, 2959. https://doi.org/10.3389/fpsyg.2020.510029.

Felton, W.; Jackson, R. Presence : A Review. *International Journal of Human–Computer Interaction*, **2022**, *38*, 1-18. https://doi.org/10.1080/10447318.2021.1921368.

Food Experiment Turku December **2021**: https://youtu.be/lxhuD7qbbwU

Fröber, K.; Thomaschke, R. In the dark cube: Movie theater context enhances the valuation and aesthetic experience of watching films. *Psychol. Aesthet. Creat. Arts.* **2021**, *15*, 528–544. https://doi.org/10.1037/aca0000295.

Galmarini, M. V., Paz, R. S., Choquehuanca, D. E., Zamora, M. C., & Mesz, B. (2021). Impact of music on the dynamic perception of coffee and evoked emotions evaluated by temporal dominance of sensations (TDS) and emotions (TDE). *Food Research International*, 150, 110795.

Hosoya, G.. The artwork and the beholder: A probabilistic model for the joint scaling of persons and objects. *Psychol. Aesthet. Creat. Arts.* 2020, *14*, 224–236. https://doi.org/10.1037/aca0000239.

Keller, D., Otero, N., & Costalonga, L. (2015a). Aesthetic heuristics in ubimus. *Electronic Visualisation and the Arts* (EVA 2015), 64-71.

Kahn, A. S.; Cargile, A. C. Immersive and Interactive Awe: Evoking Awe via Presence in Virtual Reality and Online Videos to Prompt Prosocial Behavior. *Hum. Commun. Res.* **2021**, *47*, 387–417. https://doi.org/10.1093/hcr/hqab007.

Keller, D., Otero, N., Lazzarini, V., Pimenta, M. S., de Lima, M. H., Johann, M., & Costalonga, L. (2015b). Interaction aesthetics and ubiquitous music. *Creativity in the digital age*, 91-105.

Keller, D., Alcántara-Silva, T. R., & Mesz, B. (2021, September). Dossiê Ubimus na Revista Vórtex: Ubimus, Gastrossônica e Bem-estar. In 11th Workshop on Ubiquitous Music (UbiMus 2021). g-ubimus.

Keller, D., Simurra, I., Messina, M., Neiva, T., Tedesco, S., & Mesz, B. (2022). Domestic ubimus. EAI Endorsed Transactions on Creative Technologies, 9(30).

Lee, K. M. Presence, Explicated. *Commun. Theory* **2004**, *14*, 27-50. https://doi.org/10.1111/j.1468-2885.2004.tb00302.x.

Makransky, G.; Lilleholt, L.; Aaby, A. Development and validation of the Multimodal Presence Scale for virtual reality environments: A confirmatory factor analysis and item response theory approach. *Comput. Hum. Behav.* **2017**, *72*, 276-285. https://doi.org/10.1016/j.chb.2017.02.066.

Marković, S.; Dorčić, J.; Rašan, D.; Bucić, B.; Blažić, M. Aesthetic Guest Experience In Restaurant: A State-Of-The-Art Review. EMAN 2021–Economics & Management: How to Cope with Disrupted Times, **2021**. 135. https://doi.org/10.31410/EMAN.S.P.2021.135.

Menninghaus, W.; Wagner, V.; Wassiliwizky, E.; Schindler, I.; Hanich, J.; Jacobsen, T.; Koelsch, S. What are aesthetic emotions? *Psychol Rev.* **2019**, *126*, 171-195. https://doi.org/10.1037/rev0000135.

Mesz B, Trevisan MA, Sigman M. 2011. The taste of music. *Perception*, 40(2), 209-219.

Mesz, B. A., Tedesco, S., & Potenza, L. (2023). Quorum Sensing y Vajilla Emocional. *Revista Vórtex*, 11(1), 1-24.

Milgram, P.; Takemura, H.; Utsumi, A.; Kishino, F. Augmented reality: a class of displays on the reality-virtuality continuum. *Proc. SPIE*, **1994**, *2351*, 282–292. https://doi.org/10.1117/12.197321.

Motoki, K.; Park, J.; Spence, C.; Velasco, C. Contextual acceptance of novel and unfamiliar foods: Insects, cultured meat, plant-based meat alternatives, and 3D printed foods. *Food Qual. Pref.* **2022**, *96*, 104368. https://doi.org/10.1016/j.foodqual.2021.104368.

Muhanna A. Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions, *Journal of King Saud University - Computer and Information Sciences*, **2015**, *27*, 344-361. https://doi.org/10.1016/j.jksuci.2014.03.023.

Nakagawa, S.. A farewell to Bonferroni: the problems of low statistical power and publication bias. *Behav. Ecol.,* **2004**, *15*, 1044 045. https://doi.org/10.1093/beheco/arh107.

Nivedhan, A.; Mielby, L. A.;Wang, Q.J. The influence of emotion-oriented extrinsic visual and auditory cues on coffee perception: a virtual reality experiment. *Proceedings of the 4th International Workshop on Multisensory Approaches to Human-Food Interaction* **2020**. https://doi.org/10.1145/3395035.3425646.

Reinoso-Carvalho, F., Gunn, L., Molina, G., Narumi, T., Spence, C., Suzuki, Y., ... & Wagemans, J. (2020). A sprinkle of emotions vs a pinch of crossmodality: Towards globally meaningful sonic seasoning strategies for enhanced multisensory tasting experiences. *Journal of Business Research*, 117, 389-399.

Riva, G.; Mantovani, F.; Capideville, C.; Preziosa, A.; Morganti, F.; Villani, D.; Gaggioli, A.; Botella, C.; Alcañiz Raya, M. Affective interactions using virtual reality: the link between presence and emotions. *Cyberpsychol. Behav.*, **2007***, 10*, 45–56. https://doi.org/10.1089/cpb.2006.9993.

Rob Verf **2021**: Inside the Cheese: https://youtu.be/695jl1DyeAA Rob Verf **2021B**: The expressive life inside my plastic body: https://youtu.be/_e5LwMbuSOU Rosales, E. (2023). Taste This Score. *Revista Vórtex,* 11(1), 1-14.

Schindler, I.; Hosoya, G.; Menninghaus, W.; Beermann, U.; Wagner, V.; Eid, M.; Scherer, K. Measuring aesthetic emotions: A review of the literature and a new assessment tool. *,* **2017**,*12*, e0178899. https://doi.org/10.1371/journal.pone.0178899

Schouteten, J. J., De Steur, H., De Pelsmaeker, S., Lagast, S., De Bourdeaudhuij, I., & Gellynck, X. (2015). Impact of health labels on flavor perception and emotional profiling: A consumer study on cheese. *Nutrients*, 7(12), 10251-10268.

Slater, M.; Wilbur, S. A Framework for Immersive Virtual Environments (FIVE): speculations on the role of presence in virtual environments. *Journal Presence: Teleoperators and Virtual Environments archive*, **1997**, *6*, 603–616. https://doi.org/10.1162/pres.1997.6.6.603.

Spence C. 2011. Crossmodal correspondences: A tutorial review. *Attention, Perception, & Psychophysics*, 73(4), 971-995.

Spence, C. Temperature-based crossmodal correspondences: Causes and consequences. *Multisensory res.*, **2020**, *33*, 645-682. https://doi.org/10.1163/22134808-20191494
Spence, C. Sonic Seasoning and Other Multisensory Influences on the Coffee Drinking Experience. *Front. Comput. Sci.*, **2021**, *3*, . https://doi.org/10.3389/fcomp.2021.644054.

Spence, C. (2023). Digitally enhancing tasting experiences. *International journal of gastronomy and food science*, 100695. https://doi.org/10.1016/j.ijgfs.2023.100695.

Spence, C.; Gallace, A. Multisensory design: Reaching out to touch the consumer. *Psychol Mark.*, **2011**, *28*, 267–308. https://doi.org/10.1002/mar.20392.

Spence, C., & Wang, Q. (2015). Wine and music (II): can you taste the music? Modulating the experience of wine through music and sound. Flavour, 4, 1-14.

Wang, Q.J.; Escobar, F.; Alves Da Mota, P.; Velasco, C. Getting started with virtual reality for sensory and consumer science: Current practices and future perspectives. *Int. Food Res. J.*, **2021**, *145*,110410. https://doi.org/10.1016/j.foodres.2021.110410.

Wang, Q.J.; Meyer R.; Waters S.; Zendle D. A Dash of Virtual Milk: Altering Product Color in Virtual Reality Influences Flavor Perception of Cold-Brew Coffee. *Front. Psychol.*, **2020,** *11*, https://doi.org/10.3389/fpsyg.2020.595788.

Wang, Q. J., & Spence, C. (2017). The role of pitch and tempo in sound-temperature crossmodal correspondences. *Multisensory research*, 30(3-5), 307-320.

Webinar in Turku, Finland 10.12.2021 https://www.youtube.com/watch?v=ngd7fTbsl1w&t=5628s.

Wilkinson, M.; Brantley, S.; Feng, J. A Mini Review of Presence and Immersion in Virtual Reality. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, **2021**, *65*, 1099-1103. https://doi.org/10.1177/1071181321651148.

Musical stuff, technologically convergent and disruptive factors

Marcello Messina[1,2,3,4], Brendah Freitas[2,4] , Ivan Simurra[2,4], Carlos Gómez[2,3], Luzilei Aliel[4], Damián Keller[2,3,4]

[1] Southern Federal University, Rostov-on-Don, Russia
[2] NAP, Federal University of Acre, Rio Branco, Brazil
[3] NAP, Federal University of Paraíba, João Pessoa, Brazil
[4] Ubiquitous Music Group

messina@sfedu.ru, dkeller@ccrma.stanford.edu

Abstract. *We introduce musical stuff as a potentially unifying place-holder for post-2020 musical practices enabled by ubiquitous technologies. Musical stuff incorporates a flexible approach to the design, implementation and deployment of networked resources, inheriting key characteristics from ubimus frameworks. We discuss the material requirements for aesthetically pliable designs and analyze the possibilities and problems of adopting 'stuff' as a working denomination. Our theoretical discussion is exemplified with case scenarios tied to the various dimensions proposed. We highlight the challenges faced by ubimus due to the disruptive tendencies of fast or socially brittle technological change.*

Resumo. *Apresentamos o bagulho musical como um espaço potencialmente unificador para as práticas musicais pós-2020 viabilizadas por tecnologias computacionais ubíquas. O material musical incorpora uma abordagem flexível para o design, a implementação e a implantação de recursos computacionais, herdando as principais características das práticas ubimus. Discutimos os requisitos materiais para o desenvolvimento de designs esteticamente flexíveis e analisamos as possibilidades e os problemas de adotar o 'bagulho' como conceito. Nossa discussão teórica é exemplificada com cenários vinculados às propostas de implementação. Destacamos os desafios enfrentados pela pesquisa ubimus a partir das tendências disruptivas das mudanças tecnológicas rápidas ou com embasamento social frágil.*

1. Post-2020 disruptions of musical working units

The pressures motivating the emergence of ubiquitous music frameworks started to build up in the early nineties and erupted by the turn of the century. A quick glance points to network technologies, embedded computing and the popularization of mobile devices as the main drivers of ubimus initiatives. Nevertheless, mobility, connectivity and alternative strategies to musical interaction were present (although not widely adopted) since the mid 1980s.[1] Therefore, during a first wave of ubimus proposals the musical-interaction concepts did not depart entirely from the hegemonic acoustic-instrumental paradigm despite the fact that new approaches were being tested in the field [Pimenta et al. 2014].

The weight of instrumental thinking can be gauged by looking at the targets of two perspectives to musical interaction that may be interpreted as being complementary to ubimus: Networked Music Performance and Telematic Art [Ascott 1984; Lazzaro and

[1] Musical technologies were networkable since the introduction of the MIDI standard in 1983 (https://www.midi.org/specifications).

Wawrzynek 2001]. Both of them employ the acoustic-instrumental chamber ensemble (or its expanded form, the orchestra) as the ideal mold for musical-interaction design. An ultimate technical target is to achieve a "realtime" performance free of delay and jitter. A frequently used pattern of organization is the master-slave hierarchy, typically represented by a centralized scheduling system that attempts to emulate the relationship of an orchestral conductor with an instrumentalist.[2]

An example of a problematic notion tied to the deployment of ubiquitous technologies is the concept of *musical things*. This concept appeared in various projects during the 2010s and eventually was incorporated as a central component of the Internet of Musical Things [Keller and Lazzarini 2017; Ribeiro Netto et al. 2015; Roy et al. 2018; Turchet et al. 2018; Zawacki and Johann 2014]. Since the revision of proposals laid out by Turchet et al. (2018), multiple close derivatives of the same concept have been proposed, including the Internet of Media Things, the Internet of Sounds, the Web of Audio Things, the Internet of Everything, the Internet of Anything,[3] etcetera etcetera. All these proposals can be gathered under the rubric of Internet of X Things or IoXT (to create yet another acronym!).[4] These, of course, deserve individual treatment and might eventually coalesce into a useful body of knowledge for music-making. For the moment, we will focus exclusively on the concept of *things* and its implications for post-2020 musical practices. Thus, we will not engage with discussions of whether urban sounds belong to the musical realm or not since this issue is already settled by the ongoing practice of soundscape composition [Schafer 1977]. In any case, a note of caution is necessary. Within the ubimus frameworks, music may be defined as *organized sonic information* [Keller et al. 2021]. Thus, musical experiences include symbolic musical data. But they are definitely NOT just that. Ubimus frameworks engage with behavioral ecologies and multisensory (or multimodal) ecologies [Keller and Lazzarini 2017]. Consequently, limiting music-making to handling symbolic data or to producing sound is unnecessarily restrictive -although sound production is something fundamental - and implies an inaccurate depiction of musical experience.[5]

Having provided the context and the motivations to introduce a new working denomination for musical-interaction design, we now turn to the specific caveats of the extant proposals that deal with musical things as building blocks of socio-technological frameworks for music-making. We take as a given the existence of a *musical internet* which provides a platform for the deployment of resources to support various forms of musical practice. The remaining sections of this paper are organized as follows. Section 2 introduces a new definition of musical stuff that complements the development of second-wave ubimus frameworks. Section 3 provides examples of stuff deployments.

---

[2] The political implications of this widespread metaphor are manifold. We will address these issues in an upcoming publication on ubimus decoloniality. For instance, the conductor is pompously called Maestro -- reinforcing the notion of a unique "master" within a context of non-reciprocal relationships.

[3] https://www.computer.org/csdl/magazine/co/2014/06/mco2014060072/13rRUxC0SSm [Accessed 18.06.2023].

[4] De Haan (2015) suggests "the internet of whatever" which tangentially relates to stuff as a nonhierarchical notion of knowledge organization.

[5] What constitutes a musical experience is, of course, a long-standing research and philosophical question. Ubimus provides open concepts and methods to deal with this issue. Thus, rather than a 'theory', it may be characterized as a 'practice' or a way of dealing with musical experiences.

Section 4 summarizes the proposals and identifies major challenges to be addressed to enable the usage of stuff in field work.

2. Musical stuff

As an exercise to shake the ongoing acoustic-instrumental biases enforced by mainstream musical-interaction design, this section introduces a potentially polemic and possibly not intuitive definition of a working denomination for post-2020 music making: *musical stuff* [Messina et al. 2022a; Messina et al. 2022b].[6] From the start, we want to stress that we are not dealing with a purely functional concept, replaceable by technological "things". Our motivations, as stated in the previous section, are partially methodological and partially epistemological. We do not only target new ways of music-making. We also target new ways of thinking about artistic practice. Our approach is backed by a decade and a half of ubimus deployments, recently expanded to the realm of musicology [Maciel, Costa and Costalonga 2022; Menezes and Lopes 2022; Mesz, Tedesco and Potenza 2022].

Text box 1. A provisional four-component definition of musical stuff.

(1) A phenomenology of (2) pliable entities that enable (3) distributed creative activities, (4) deployable on the musical internet.

Musical stuff is defined as a phenomenology of pliable entities that enable distributed creative activities, deployable on the musical internet. The four components of the concept complement each other and are intimately linked to a recent definition of ubiquitous music (the study of systems encompassing human and nonhuman stakeholders that afford creative music-making by means of technological ecosystems). Thus, musical stuff may be conceptualized as the material grounding necessary to deploy ubimus activities or as the negotiated field of material constraints and opportunities furnished by infrastructure.[7]

(1) Phenomenology "asks us to be aware of the 'what' that is 'around'" (Ahmed 2007: 151) rather than focussing on pre-packaged, given ontologies. "Things" do not come as fixed entities, provided with a set of ideal, quasi-metaphysical properties. They are rather the result of the cognitive segmentation of existing reality ("stuff"). This fixed segmentation is perhaps counterproductive in the context of the musical internet.

(2) Pliable entities refer to the material resources featured in ubimus ecosystems. The profiles of these resources are constrained by the laws of physics, by the biological evolutionary pressures and by the dynamics of social interactions. Hence, the entities that constitute stuff are not just digital information that can be acted upon or manipulated at will. They are the result of a history of previous interactions that tends to fix their qualities, behaviors and relational properties, shaping their potential to interact with other stakeholders and to generate new creative byproducts and experiences.

---

[6] Adopted translations include *bagulho* (Portuguese), *cachivache* (Spanish) and *roba* (Italian).

[7] When we talk about infrastructure, we do not abide by the inherited materializations of social relationships typically exemplified by instrumental practice and instrumental ensembles. For us, infrastructure is negotiation, impacted by biological, economic and social pressures shaped by materially grounded and socially situated cognition.

(3) Distributed creative activities encompass the actions and processes afforded by musical stuff. These activities may be carried out in a shared space or they may encompass multiple settings. The locations of the resources and stakeholders will influence the required material support, establishing at least three possible categories of interaction: synchronicity, asynchronicity and quasi-synchronicity [Messina et al. 2022b]. Several ubimus projects have explored quasi-synchronous activities, often implying a relaxed approach to temporalities (including strongly heterogeneous temporal behaviors) and departing from the master-slave model enforced by Networked Music Performance.[8]

(4) To be deployable on the musical internet entails the possibility to use networked resources. The musical internet is defined as a set of aggregated resources shaped by ongoing negotiations among multiple stakeholders. Thus, accessibility and reliability cannot be taken for granted. Effective deployments of ubimus ecosystems depend on the ability of the stakeholders to adapt their behaviors to the local conditions by relying on the pliability of the resources. This push and pull among stakeholders and resources enforces their resilience (when sustainable strategies are adopted) or their brittleness (when legacy resources are not aligned with the current technological requirements) of the ubimus ecosystems. This is an emerging area of study which will demand an important effort to unveil major influential factors and possible strategies for future parameterization, assessment and possibly standardization.[9]

3. Some Examples of musical stuff

This section provides examples of applications. As it is the case with other ubimus concepts, the limitations and the power of stuff are gauged through actual deployments, data collection and analysis of feasibility. We provide pointers on how to engage with these tasks when aiming at deployments in the field.

3.1. Ntrallazu and Radical ASC

Simurra et al. (2022) present two ubimus artistic projects based on semantic strategies that address the three issues identified above. Namely, they entail the use of semantic epimusical resources[10] [Keller, Messina and Oliveira 2020], they employ local referents and resources to engage with aspects of everyday creativity and they adopt distributed decision-making procedures to avoid hierarchically oriented social exchanges. These methods share characteristics that justify their treatment as a block: They constitute a family of approaches based on the use of semantics; They avoid the hierarchical division of labor of the acoustic-instrumental paradigm by proposing open-ended negotiations among the stakeholders; And they play with the listeners' expectations by shifting the meanings of the referents.

[8] Despite the objection of a reviewer, we chose to keep the critique of the master-slave model. this terminology is not uncommon and hints to a colonial vision of infrastructure.
[9] A useful thread is provided by the techniques employed in archaeological ubimus [Lazzarini and Keller 2021].
[10] In contrast with extra-musical resources, epimusical resources impact the sonic outcome directly.

Ntrallazzu (Messina and Aliel 2019; Messina and Feichas 2020) is the title of a cycle of works based on interactive live scores that reflects on the multifaceted concepts of liveness and interaction both from a creative point of view and as a critical philosophical perspective. Both the live score and the electronics run on Max[11]. The pieces that compose the cycle are between 5 and 6-minutes long and rely on a projected score that interacts in real time with the materials played by the performer(s). While one of the performers is playing, the sonic output is processed by a patch that generates both the electronic sounds and the score. This score is usually (but not always) performed by a second player.

Ntrallazzu 4 and *Ntrallazzu 6*[12] benefit respectively from indirect and direct applications of creative semantic anchoring (ASC – Simurra et al. 2022): at the intersection between these two different approaches, lies a general commitment to the theoretical premises of the piece. If, in *Ntrallazzu 6*, these premises radically become the same textual/sonic material upon which the piece itself is predicated, in *Ntrallazzu* 4 they inform verbal negotiations that end up shifting the very functioning of the piece. For both cases, we propose the rubric of "Radical ASC" to describe the operative performativity of semantics in shaping the musical and epimusical characteristics of the creative processes, to an extent that transcends the pertinence of the "traditionally notated" instructions.

Finally, Simurra et al. (2022) propose extreme cases of Radical ASC where the agents facilitate the production of creative outcomes whereby the linguistic sign (as related to meaning) becomes the most important or the only pertinent musical parameter. Spoken-word pieces conceived as works of experimental music may be among these extreme cases: as an example, Lauren Redhead's album *solo speaking* (2016),[13] a compilation of pieces with articulated verbal language as the primary musical element involved in the creative interaction. Here semantics supersedes phonetics in determining creative decisions, both from a compositional and from a performance-oriented point of view. As suggested by Simurra et al. (2022), Radical ASC may be seen as promoting a shift within musical interaction, from sonic outcomes as targets to the crafting of meanings that may ignore the specificities of aural encounters and interactions.

3.2. The Max paradigm and Viscosity-fluidity

Viscosity describes a resource's resistance to change, dependent on the procedural and material investments needed to change a design decision [Bellingham et al. 2014]. *Repetition viscosity* is applied to cases in which several actions are necessary to achieve a single goal. *Knock-on viscosity* refers to remedial actions required to restore a desired operation. Bellingham et al. (2014) point to a close relationship between *viscosity* and *premature commitment*. Premature commitment, also described as *early domain restriction* [Lima et al. 2012] refers to decisions taken before the necessary information is available, which may impact *layout* or *connections* [Green and Petre 1996].

[11] In this case we refer to the software, not the paradigm discussed by Puckette (2002).
[12] Available at <https://doi.org/10.5281/zenodo.4425834> [Accessed 14.04.2023]
[13] Available at:<https://laurenredhead.bandcamp.com/album/solo-speaking> [Accessed 19.06.2023]

Consider the design of visual languages for music making. Most visual languages for composition force commitment to connections by adopting a graph-like, data-driven model (a notable example is Max paradigm[14] – Puckette 2002). Given the reliance on visual entities, the complexity of the algorithm is limited by the visibility of the objects and the connections, which in turn are constrained by the available screen space and by the level of detail enforced by the representation. To make connections among resources implies to exclude other possible connections. Hence, rather than deferring the important decisions to the latest stages of the creative cycle, the stakeholders are induced to commit to a specific musical approach from the start. This is a subtle form of *early domain restriction* because it only becomes explicit through the analysis of usage. Neither the musical outcome nor the functionality of the language point to this characteristic.

Another example of high viscosity can be observed in the common-practice-notation model adopted by most sequencers. Musical events are displayed from left to right and are rendered sequentially. This type of display works well for performance-oriented systems because the spatial position of the symbols tend to be aligned with the temporal position of the events. Contrastingly, by using *out-of-time* techniques the access to the events is not tied to the temporal order of the resources. Musical data can be manipulated freely depending on the creative needs of the stakeholders. Decisions on resources belonging to the initial temporal frames of the creative product do not necessarily impact the decisions on resources employed in subsequent temporal frames. Hence, the flow of creative decisions tends to be determined by the needs of the participants rather than by the visual representations.

3.3. Kiwi and Territoriality

Metaphors of geopolitical control and territorialized desire are not uncommon within creative collaboration, even in the case of distributed and asynchronous interaction [Messina et al. 2019]. The complex and multiple activity of users across time and space over the same resources is likely to generate some sort of conflict, be it metaphorical or concrete. Drawing upon previous ubimus experiences, Kramann (2020) argues that this type of conflictuality may prove to be a substantial obstacle in the context of distributed creativity. We maintain that some degree of territoriality and conflictuality is, by definition, a necessary part of human (and non-human) interaction. However, we consider Kramann's argument extremely relevant, particularly when conflict and territoriality may escalate into situations of extreme imbalance in terms of access to resources. We argue that the forced objectification and tokenization that we address in this paper is one of the elements that permits such escalation. In general terms, ubimus reflections on territoriality are complementary to the fostering of ecologically grounded creative practices [Keller and Lazzarini 2017].

[14] The Max paradigm includes three programs: Max/MSP, Jmax, and Pd, which can be considered as different implementations of the same idea. This paradigm "can be described as a way of combining pre-designed building blocks into configurations useful for real-time computer music performance." [Puckette 2002: 31].

Messina et al. (2019) report on an ubimus experience with the software Kiwi[15] that targets synchronous and collaborative live-patching by participants situated across continents. The creative products consisted of collaborative patches without genealogical tracks. In the other words, the activity makes it very hard to trace who created each object, who established each connection between the objects or who commented on a patch. Also, the patches are available and liable to modifications by a succession of multiple users. Subverting the logic of scarcity and social tags linked to "property", this open, collaborative and non-hierarchical, working method furnishes an example of distributed creation based on the usage of musical stuff. In particular, territoriality is a relational property that is highlighted in this experience. Another aspect that emerges from the analysis of the interactions is the property of rivalry [Keller 2014].

3.4. *pulse2357* and Monetization

Taking as a starting point the ubimus principle of distributed creativity, Kramann (2020) devised *pulse2357* as a "board game with an inherent correlation to music" (2020: 24). Implemented and disseminated online as a free Android application, the game becomes a tool for composition (2020: 29). Featuring a limited range of actions, the intention is to quickly familiarize untrained users with selected aspects of musical creation. As a tool for algorithmic music-making, pulse2357 defies the reification. One aspect is the form of dissemination of the board game, distributed as a free mobile application — though sharing free software on Android's Play Store is not the exclusive prerogative of Kramann's work. Secondly, the musical material produced while playing pulse2357 does not constitute a creative "work". The outcomes exist ephemerally in the form of volatile game practices. For instance, we (happily) fail to envisage the material resulting from pulse2357 games as tokenizable into NFTs and sellable as — to quote Ethereum's corporate rhetoric — "the real thing".[16] Finally, the underlying design strategy proposed by Kramann, Arithmetic Operator Grammar, is an example of ubimus-oriented computational thinking. The creative outcomes are constrained by the sonic relationships established by the generative algorithm. These relationships are dynamic and open, thus they are not tied to a predefined "genre". Kramann's approach provides a good example of an aesthetic practice compatible with the concept of musical stuff that adopts a self-determined musical logic.

---

[15] A brief presentation of this software together with the download links for its installation can be found at: http://musicoll.github.io/Kiwi/#page-top [Accessed 18.06.2023]

[16] From the Ethereum description of NFTs (Non-Fungible Tokens): "The copy/paste problem. Naysayers often bring up the fact that NFTs 'are dumb' usually alongside a picture of them screenshotting an NFT artwork. 'Look, now I have that image for free!' they say smugly. Well, yes. But does googling an image of Picasso's Guernica make you the proud new owner of a multi-million dollar piece of art history? Ultimately owning the real thing is as valuable as the market makes it. The more a piece of content is screen-grabbed, shared, and generally used the more value it gains. Owning the verifiably real thing will always have more value than not" <https://ethereum.org/en/nft/> [Accessed 13.04.2023]. Our critique of NFTs and of the logic of the "real thing" professed in Ethereum's corporate rhetoric is available in Messina et al. (2022b).

4. Conclusions and next steps

Post-2020 musical practices have brought to the forefront several preoccupations of ubimus research that date back to the first-wave ubimus initiatives. How can ubimus designs foster collaborative, creative and socially meaningful interactions when dealing with legacy infrastructure? It seems that rather than taking the musical internet as a tabula rasa to force an alignment with colocated instrumental performance, ubimus strategies rely on a process of push and pull among stakeholders and resources. These negotiations imply the existence of a knowledge pool linked to the locally available resources that on the one hand constrains the potential designs and on the other hand encourages the stakeholders to search for adaptations to address specific needs. These design threads indicate various emerging issues in ubimus practice. One of them is the lack of a viable common ground for the expansive aesthetic perspectives fostered by ubimus frameworks. We have laid out the notion of musical stuff as a pliable but consistent construct for ubimus design, deployment and evaluation, pointing to its potential incorporation to emerging proposals in ubimus-oriented musicology.

We understand the relationship between musical stuff and semantics in terms of a twofold operationality. On the one hand, semantics is considered in terms of the complex manipulations with which human beings collectively assign meaning to words and segment reality to identify different "things": this hints at the idea that we invent or identify things by means of language, and that things are portions of reality that are segmented from other entities mainly on account of the extant human needs. Therefore, imagining "an internet of things" would hardly go beyond a mere mapping of material reality onto digital objects — this limitation underlines the need to put forward a case for an internet of (musical) stuff. On the other hand, in approaches such as Radical ASC, musical stuff can itself convey semantic content, or in extreme cases semantic content can itself become musical stuff. These extreme proposals also have the potential of encouraging a disavowal of "things" in favor of pulverized and ever-changing "stuff", in the form of dynamic processes that unfold as cross modal phenomena.[17]

Our treatment of stuff focuses on its intrinsic qualities, including semantics and territoriality — or, also, (de-)territorialisation (cf. Costa 2018; Messina, Costa and Scarassatti 2020) — also calling for further developments of design dimensions to be addressed by future studies. Aesthetic pliability is highlighted as a valued asset, setting ubimus apart from the post-2020 artistic approaches based on the enforcement of acoustic-instrumental thinking. The examples presented unveil both convergent and disruptive technological factors which may open new opportunities to shape post-2020 creative musical endeavors. Thus, musical stuff may constitute a simple but not simplistic construct that inherits practical ubimus knowledge unveiling new paths toward aesthetic diversity.

[17] These limits of semantics are starting to be explored in gastrosonics (See Proceedings of the Ubimus Symposium 2022).

5. References

Ahmed, S. (2007). A phenomenology of whiteness. Feminist Theory.8(2). https://doi.org/10.1177/1464700107078139

Ascott, R. (1984). Art and Telematics: towards a network consciousness. *Art Telecommunication.* Vancouver: The Western Front, 25-67.

Bellingham, M., Holland, S. & Mulholland, P. (2014). *An analysis of algorithmic composition interaction design with reference to Cognitive Dimensions* (Technical Report 2014/ 04). The Open University.

Costa, V. F. (2018). Comentários sobre a possibilidade de autopoiese da obra musical e sobre o performer como seu componente sistêmico. *DEBATES — Cadernos do Programa de Pós-Graduação em Música*, (21).

De Haan, G. (2015, July). HCI Design Methods: where next? from user-centred to creative design and beyond. In *Proceedings of the European Conference on Cognitive Ergonomics 2015* (pp. 1-8).

Green, T. R. & Petre, M. (1996). Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *Journal of Visual Languages and Computing* 7, 131–174.

Keller, D. (2014). Characterizing resources in ubimus research: Volatility and rivalry. Proceedings of the V Workshop in Ubiquitous Music (V UbiMus). http://compmus.ime.usp.br/ubimus2014/

Keller, D., & Lazzarini, V. (2017). Ecologically grounded creative practices in ubiquitous music. *Organised Sound*, *22*(1), 61-72.

Keller, D., Messina, M., & Oliveira, F. Z. (2020). Second wave ubiquitous music. *Journal of Digital Media & Interaction*, 3(5), 5-20.

Keller, D., Aliel, L., Filho, M. C., & Costalonga, L. (2021). Toward Ubimus Philosophical Frameworks. *Open Philosophy*, 4(1), 353-371.

Kramann, G. (2020, November). Of renouncing to do something grandiose. In Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020). g-ubimus.

Lazzarini, V., & Keller, D. (2021, September). Towards a Ubimus Archaeology. In Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020). Porto, Portugal: Ubiquitous Music Group.

Lazzaro, J., & Wawrzynek, J. (2001, January). A case for network musical performance. In *Proceedings Of The 11th International Workshop On Network And Operating Systems Support For Digital Audio And Video* (pp. 157-166), New York, NY.

Lima, M. H., Keller, D. Pimenta, M. S., Lazzarini, V., & Miletto, E. M. (2012). Creativity-centred design for ubiquitous musical activities: Two case studies. *Journal of Music, Technology & Education* **5** (2). 192-222. (DOI: 10.1386/jmte.5.2.195_1)

Maciel, C., J., R., Costa, F. A., Costalonga, L., L. (2022). *Complexidade e Interatividade como características ubíquas na música de Jacob Collier.* Anais do

Simpósio de Música Ubíqua 2022 (Ubimus 2022). 94-101. https://ubimus2022.unespar.edu.br/

Menezes, J., Lopes, E. (2022). Proceedings of the Ubiquitous Music Symposium (UbiMus 2022). 36-59.

Messina, M., & Aliel, L. (2019). Ubiquitous music, Gelassenheit and the metaphysics of presence: Hijacking the live score piece *Ntrallazzu* 4. In *CMMR 2019* (pp. 685-695).

Messina, M., Svidzinski, J., de Menezes Bezerra, D., & Ferreira, D. (2019). Live patching and remote interaction: A practice-based, intercontinental approach to kiwi. In *14th International Symposium on Computer Music Multidisciplinary Research*.

Messina, M., & Feichas, L. V. (2020). Interactivity/interpassivity and presence/absence in the Ntrallazzu cycle. In *Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020)*. g-ubimus.

Messina, M., Costa, V. F., Scarassatti. (2020). Cartridge Music in the Quarantine: Presence, Absence, Contingency Setups and (De-)territorialised Performances. INSAN Journal. 28-45. DOI:10.51191/issn.2637-1898.2020.3.5.28

Messina, M., Célio Filho, M., Mejía, C. M. G., Keller, D., Aliel, L., & Simurra, I. (2022a). A Internet do Bagulho Musical (Internet of Musical Stuff)-IoMuSt. In *Proceedings of the Ubiquitous Music Symposium (ubimus2022)* (pp. 85-93). g-ubimus.

Messina, M., Keller, D., Aliel, L., Gomez, C., Célio Filho, M., & Simurra, I. (2022b) The Internet of Musical Stuff (IoMuSt): ubimus perspectives on artificial scarcity, virtual communities and (de) objectification. *INSAM Journal of Contemporary Music, Art and Technology*, (9), 26-50.

Mesz, B. Tedesco, S. Potenza, L. (2022). Quorum Sensing. Proceedings of the Ubiquitous Music Symposium (UbiMus 2022). 77-84.

Simurra, I., Messina, M., Aliel, L., & Keller, D. (2022). Radical creative semantic anchoring: creative-action metaphors and timbral interaction. Organised Sound, 1-14.

Pimenta, M. S., Miletto, E. M., Keller, D., Flores, L. V., Teste, G, G. (2014) Technological Support for Online Communities Focusing on Music Creation: Adopting Collaborating, Flexibility, and Multiculturality, From Brazilian Creativity Styles. IG Global. DOI: 10.4018/978-1-4666-5942-1.ch038

Puckette, M. (2002). *Max at seventeen*. Computer Music Journal. 26(4). 31-43.

Ribeiro Netto, A. Castheloge, L., Oliose, A., Mateus, A., Costalonga, L., Coura, D. (2015) Árvore das memórias: Instalação Multimídia Interativa. XV Simpósio Brasileiro de Computação Musical. 76-83.

Schafer, R. M. (1977). *The tuning of the world.* New York, NY: Knopf.

Roy, S. Sarkar, D., Hati, S., Debashis, D. (2018). Internet of Music Things: An edge computing paradigm for opportunistic crowdsensing. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-018-2511-6

Turchet, L., Fischione, C., Essl, G., Keller, D., & Barthet, M. (2018). *Internet of musical things: Vision and challenges.* IEEE access, 6, 61994-62017.

Zawacki, L. F., & Johann, M. O. (2014). A Prospective Analysis of Analog Audio Recording with Web Servers. *Cadernos De Informática*, *8*(2), 20–33. Recuperado de https://seer.ufrgs.br/index.php/cadernosdeinformatica/article/view/v8n2p20-33

From Audible to Visible Ecosystems: Emergence by Modeling and the Metastable Equilibrium Problem

Ricardo Thomasi

School of Communication and Arts
University of Sro Paulo (USP/CNPq)

ricardothomasi@usp.br

Abstract. *Ecologically-grounded models for artistic performance have changed traditional perspectives on interactive systems. Spatially-extended systems exhibit spatiotemporal dynamics favoring the observation and experimentation with emergent structures. Theoretically supported by Gilbert Simondon's Allagmatic theory, we propose the idea of emergence by modeling comprising an operational approach and theoretical principle for musical structuring of interactive systems. In this paper, we present a brief comparison between the audible and visible ecosystem models developed, corroborating the metastability hypothesis as the core of an ecosystemic approach for modeling interactive systems.*

1. Introduction

New perspectives on interactive systems for music and sound art have appeared in ecologically-grounded models offering alternative paths faced with dualistic paradigms [Di Scipio, 2003; Keller and Capasso, 2006; Connors, 2017]. According to Keller and Lazarini (2017), “it incorporates place as a creativity factor highlighting the interaction with the environment as one of the central aspects of the creative process”. In that regard, we present a structural thinking called *emergence by modeling* as result of an in-depth study of recursive systems that are coupled structurally with their environment, initially inspired by Di Scipio’s AESI project [Di Scipio, 2003]. It is a particular case of artificial ecosystems [Rand, 1998] supported by spatially-extended systems [Chemo, 2019] providing an experimental territory for this investigation in the sense of artistic research purposes [Assis, 2018; Rheinberger, 2011]. We started implementing the visible ecosystems model as a contrast method to the audible ecosystems (Figure 1). The former outlined feedback possibilities beyond microphonies − Larsen effect −, obviously, as the model is based on video feedback structurally coupled to the optical medium. On the one hand, it seemed to us to be a strategy for reaching other musical applications, since the exchange of information does not take place in the audible medium, thus leaving the sound dimension to be explored by conventional acoustic and electronic instruments. On the other hand, methodological issues appeared when the theoretical model of the audible ecosystems domain, based on acoustic-digital feedback-loop, was applied to the visible domain, based on optical-digital feedback-loop, exposing the essential role of the metastability

concept in this theoretical framework. The following sessions will present theoretical aspects of the metastability and its role in the emergence by modeling model; and a comparative analysis between the audible and visual implementations highlighting the hypothesis of metastability as the core of an ecosystemic approach for interactive systems modeling.

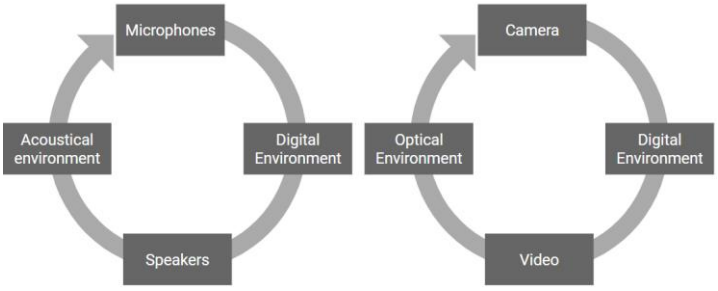


Figure 1. Visible (right) and audible (left) ecosystems feedback-loop signal flowchart. Source: the author.

2. Paradigmatic Challenges

The notion of spatially extended systems come from models of dynamic systems [Chembo, 2019], such as the cellular automaton [Vanag, 1999] and the coupled logistic map [Lloyd, 1995], in which the organization of the system depends on spatial conditions − represented probabilistically in the case of these models. The idea of artificial ecologies is a generalization of probabilistic cellular automata, in which physical space is represented by a two-dimensional LxL lattice of U sites [Rand, 1994], such as the resource-predator-prey model and the Models SugarScape [Wilensky; Rand, 2015]. In the 1980s, Crutchfield (1984, 1988) used video feedback − positioning a camera in front of a television monitor − as a "kind of spatio-temporal analog computer" capable of quickly simulating two-dimensional automatons. According to him, "studying the dynamics of this simulator is also beginning to understand a series of other problems in the theory of dynamic systems, iterative image processing, cellular automata and biological morphogenesis, for example" [Crutchfield, 1984]. Video feedback is a dissipative dynamical system, meaning energy flows through the system and is lost in microscopic degrees of freedom. "This property limits the range of possible behaviors. Starting from many different initial states, after a long period of time, the evolution of the system will occupy a relatively small region of the state space, which is the system's attractor" [Crutchfield, 1984].

However, they are two different examples of spatially-extended systems. As the first is restricted to a simulated space − in this case, the digital medium −, the second consists of a hybrid system that couple spaces of different natures by exchanging information between analog-electronic and optical mediums − in this case, referring to the processing of the camera image by the television and the screen capture method that presupposes losses, interference and distortions related to the optical medium and physical conditions of the objects, such as curvature and the television screen’s glass, for example. For this

research, spatially-extended systems following the second approach exhibit spatiotemporal dynamics favoring the observation and experimentation with emergent structures.

3. The Notion of Emergence by Modelling

We briefly define three categories of emergence that appear intertwined when musical performance is thought of as a complex system, clarifying the role of Simondon's ontogenetic approach in our proposed structural thinking. Differing from taxonomies of emergence as in Fromm (2005) and Jones (2002), in which emergent structures are classified by their level of complexity [Thomasi, 2021c], the three types shown below can vary in complexity levels. First, the *essential emergence* that underpins musical practice as a whole. Musical creation provokes sound emergence when the performance instructions are actualized by humans or machines. For instance, the sound wave propagation and their resonances as acoustic emergence. Also, the synergy among performers illustrates a kind of emergence related to interpretation and listening. They are examples of undeniable emergence belonging to the living world and the irreversible time in their many aspects. Second, the *emergence by similarity,* in which there are purely structural nexuses: one element is linked with another by structural equivalence, but they are not tied operationally in their principle [Simondon, 2020]. The gestalt paradigm may be illustrative. Distinct forms can be grouped together by external similarities and spatial organization [Lidwell and Holden, 2010]. In music, connections among composed sounds jump as emergent unity in the cognitive level, becoming autonomous as musical gesture only in an acousmatic plane [Smalley, 1996; Young, 2005]. It is when a sound event is understood as the cause or the consequence of another by juxtaposition; or by a psychoacoustic effect [Kendall, 1995]; or even when superficial similarities suggest some kind of morphological development, but they are actually disconnected events: they had independent individuation processes joined in *a posteriori* combination. We can also include the cases where elements are made by logical or symbolic organization, such as sound modeling in digital environments, re-synthesis methods and granular synthesis. The emergence comes by a structural equivalence that is only perceived by an external observer and the respective technical and epistemological objects [Simondon, 2020], i.e., in the sample level; in some preexistent rule profile; or in some symbolic programming compatibility. This is what Edmond Couchot argues as the “logic of simulation” [Couchout, 1993]. In the digital domain, the information is taken by isotropic spaces, plotting world measurements by sampling. Roughly speaking, there is no metastable condition in neutralized or simulated spaces. So, until the actualization of the information by the speakers, converting it into acoustic energy, there is no emergence but a compatible combinatorial scheme. Actually, there is no emergence by similarity that happens outside of the essential emergence limits. However, they are not what we are proposing.

In the core of this work, the *emergence by modeling* refers to a modelization of a relationship between two operations directly or mediated by a structure. Breaking the matter-form duality of the aristotelian hylomorphism,

Allagmatics presents the information as a tension that triggers a chain of transformations. Information is related to concrete realization of a signal flow; a kind of embodiment [Varela, 1995] of the information that becomes a tension of information itself, rather than a signal that works as a carrier or a container. Information triggers an energy flow among milieus with different potential energies. In this sense, structure and operation are ontological complements where the former is the result of a construction and the latter is what makes or modifies a structure [Simondon, 2020]. Emergence by modeling is essentially an operational modeling: it is *transoperative* when the relationship stands between two operations, and it is a *conversion* when between an operation and a structure. When applied in the visible ecosystems modelization, as we will see in the section 6, it comprises the organization of transoperative relationships among RGB signals and the signal conversions along the feedback-loop as well, i.e. video matrix layers to data stream and vice-versa. The light incidence that operates on the digital-signal flow provoking changes in color gradient is also an example of operations mediated by a structure, in this case, the emergent membrane. So, it is an energetic system that brings relationships to the forefront: a schematic in which the elements become sensitive to their surroundings.

4. The Concept of Modeling

While implementation refers to logical programming often restricted to software implementation [Dupuy, 1996], our modeling approach needs to involve the whole system also including the spatial relationships of the non-digital components, like microphones or camera, concerning the characteristics of respective occupied spaces – the modes of incidence of light for the camera, and the acoustic niches for the microphones [Thomasi, 2022b]. So, artificial ecosystems modeling is done in a pre-individual instance, concerning the *inter-systemic relationships* – engendering systems with certain levels of compatibility and possible to be coupled – and *intra-systemic relationships* – the energy flow responsible for the structural coupling itself that is established by potential difference of the systems and that requires a zone of tension: the metastable equilibrium. Particularly, according to Simondon, the concept of modeling is situated as an intermediate between molding and modulation [Simondon, 2020]. While molding is a finite operation, modulation is a continuous process of taking shape. Thus, modeling tends towards one extreme or the other depending on the dynamics the system exhibits: the instabilities [Schmidt, 2019]. In our modeling methodology *the possible molds and modulators are considered in a broad sense, making the arrangement of the milieu, where the interactions will take place, a creative space itself.*

5. Structural Aspects of Emergence by Modeling

The emergence phenomenon appears in different natures and it seems to be related to distinct processes of different orders, such as organization, state transitions and non-linear behavioral profiles. Such a condition makes confusing and probably impractical any straightforward approach to the emergent phenomena. Alternatively, it is possible to focus on underlying

relationships that allow emergence to occur. By looking at the micro and macro structural levels at once we arrive at a structural modeling hypothesis based on *metastable equilibrium condition, membrane-like behavior,* and *feedback topology,* corroborated by audible and visible ecosystem experiments.

5.1. The Metastable Equilibrium

Metastability represents the energetic conditions that enable emergence to happen. In thermodynamics it is a condition of matter far-from-equilibrium [Prigogine, 1977]. It can be understood as a zone of tension or critical region inferring directly in the system's behavioral profile becoming more sensitive to the dynamics of its environment. According to Prigogine, matter in equilibrium is blind, while in far-from-equilibrium it begins to see [Prigogine, 1987]. Unstable dynamic systems have fragile equilibrium points. Fluctuations over these points, that in stable systems would be irrelevant, can cause disturbances. The qualities of these fluctuations also become critical: small variations in initial conditions drive systems evolving to different directions. So, unstable dynamic systems are sensitive to initial conditions. In the Dissipative Structures Theory [Prigogine, 1977], after achieving a critical point, the far-from-equilibrium open-systems dissipate matter and energy by exchanging information with the environment, then becoming sources of new forms of order.

For Simondon, metastability is the very condition for the ontogenesis. The metastable equilibrium establishes the allagmatic relationship along the potential energy actualization. According to Simondon, "it is while the system is in a state of metastable equilibrium that it is modulable by singularities and is the theater of the process of amplification, summation and communication" [Simondon, 2020]. The internal resonance establishment is the starting point: it is a communication chain that emerges from a compatibility schema between two different energetic supports. In the audible ecosystem model it can be observed in the microphony effect where systems with different energetic support – room acoustics, microphones, speakers and digital environment – are coupled due an internal resonance – the compatibility of input and output signals that closes the feedback-loops – that is only possible after achieving a critical point – by increasing gain structure levels –, thus, entering in a condition of metastable equilibrium. It is only in the metastable equilibrium system's components become sensitive to their surroundings.

5.2 Metastability and The Open-loop

Metastability requires heterogeneous relationships which exhibit a certain compatibility schema; a dissymmetry of one in relation to another energy support. For the coupling of two systems with distinct energetic support it is necessarily the establishment of a communication chain: the internal resonance. Microphone and speaker are distinct systems but coupled by sharing the same energy flow: *a signal.* But they are not in metastable condition until the microphones begin to capture speakers' signals, closing the feedback loop [Thomasi, 2021a]. The loop formation reveals the critical point where the

microphone's behavior becomes sensitive to the speakers and to all of the intermediary components. Above this critical point, the dissipative characteristics of the system's dynamics in continuous exchange with the environment can be observed in behavioral constraints [Hooker, 2012]. What means that emergent patterns or structures formation become shaped by physical constraints present in the signal conversions and through the consequent loss of information. In the case of audible and visible ecosystems, these patterns appear as sound spectrum formations or color gradients (see Video 1 and Video 2). So, the feedback-loop has a fundamental role: first, it opens a communication chain between discrete and continuous time; it becomes an interface between the symbolic-digital world and the living world (Figure 1). Second, and as a consequence, it opens the system for complex iterations that are totally different from randomness or calculable iterated functions.

5.3 Membrane-like Behavior and Feedback Topology

According to Sauvanargues, “the membrane must not be understood as an inert limit” [Sauvanargues, 2012]. It is a spatiotemporal differentiation when metastable equilibrium creates an energetic tension that differentiates internal and external sides. In simple terms, it can be achieved when two inverse and equivalent forces regulate a signal flow. In closed systems, probably they would null themselves. But in an open system the differences or distortions of the signal embodied in the milieu are amplified throughout the very process of embodiment. By their sensitivity to fluctuations and resisting each other's forces, they never reach a common equilibrium point, working as in the play of a membrane. In this sense, *membrane-like behavior formation* is the emergence of the interactive structure itself.

Membranes have two properties: *porosity* and *polarity*. The first is responsible for allowing the incidence of differential information ensuring the metastable equilibrium. The second is responsible for defining internal and external sides and what is allowed to pass through them: it is both a consequence of the establishment of internal resonance and the source of its self-maintenance, resulting in the appearance of bubbles or structural coupling. A topology emerges with membrane polarity designating energetic evolving possibilities. *Topology means the limits of the interactive structure.* We call feedback topology since in artificial ecosystems the feedback-loop is the very tensor that supports the membrane's formation. By changing the feedback topology, we change the interaction structure and, consequently, the character of emergence observed at the macro level. Membrane and topology brings the notion of interaction to the spatial and concrete level: the spatial disposal of the objects, of the observers and modulators; physical distances between speakers and microphones or video and camera, and equipment qualities. Changes in the optical or acoustical environments cause distortions in the information flow. So, the path is not neutral on any level: the artificial ecosystem emerges from its own energetic history, its way of inhabiting [Ingold, 2015].

6. Modeling Methodology and the Problem of Metastability

6.1 Audible Ecosystems model

We delineated control strategies for performing with audible ecosystems [Thomasi, 2022b]. It was possible because we developed a feedback-loop control capable of keeping the feedback signal in a metastable equilibrium [Thomasi, 2021b]. The implementation of a digital gain control inversely proportional to the amplitude of the resonant frequency produces the emergence of a membrane as it is in constant exchange of information with the milieu, as shown in Figure 2. Whatever how many feedback-loops are created, the acoustic environment acts like a hub, converging and naturally mixing all the signals, characterizing a one-plane feedback-loop. In this sense, room acoustics acts over the feedback signal as an opposite force, shaping the signal by confronting it with its acoustic capabilities, leading to a dissipative organization [Prigogine, 2011]. Microphone and speaker placement along the room structurally couple feedback signals with respective acoustic characteristics – room modes, reverberation and absorbing surfaces – creating a range of possibilities to spectral evolving: a feedback topology [Thomasi, 2021a]. Details about the audible ecosystems model is beyond the scope of this paper and can be found in the cited publications.

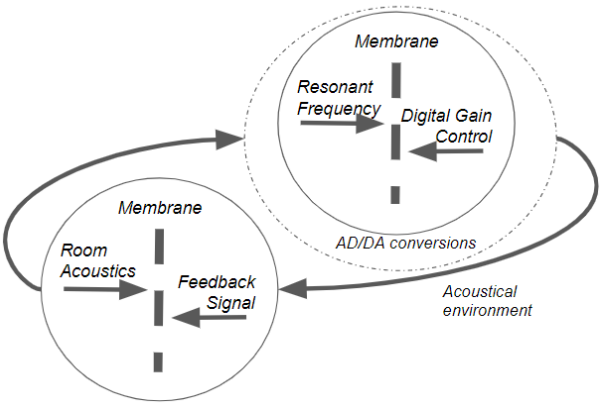


Figure 2. Illustration of inversely proportional forces whose relationship contributes to the emergence of a membrane-like behavior in the audible ecosystem model. Source: the author

6.2 From Audible to Visible Artificial Ecosystems

Video feedback has been already explored as a spatially-extended system to study dynamics and fractal theories [Crutchfield, 1988]. However, we intend with video feedback a system to investigate the principles of relational structure formation in the operative level. Audible ecosystems exhibit sounding behaviors as emergence from its digital-analog-acoustic structural coupling. Similarly, visible ecosystems show traces of its interactive structure through color gradient dynamics due to the RGB feedback-loop mapping. Despite the similarity of wave propagation in the optical and acoustical environment and between audible and visible models, *optical-digital and acoustical-digital loops are faced with different implementation problem*s. First, concerning the mapping of three

feedback-signals of the RGB planes in the visible model, rather than the one-plane feedback-loop in the audible ecosystem. The same problem led Cruchfield to start with monochromatic video feedback [Cruchfield, 1984]. Since our research interest is upon the traces left by the interactions, it is not only a mapping issue but a problem of energy flow engendering. Second, concerning the analogical nexus of the optical-digital loop. Microphone's membrane movement is analogically mapped to the speaker's membrane movement through an equivalent acoustic-electrical conversion [Eargle, 2005]. It is a compatibility schema already materialized in the original functions of the audio equipment that enables the microphone-speaker coupling responsible for the microphony effect – the Larsen Effect. In the sound context, positive feedback signal increases to the system's limit unless it is digitally controlled or nulled by an external event or another coupled system. In contrast, digital image capturing is made by analogical photosensor gradient coupled with color filters and a chipset that translates light incidence in each photosensor into electric energy, then mapping to the color planes [Filho and Neto, 1999]. The photosensor does not make a conversion, as the acoustic and electric operations mediated by the microphone's membrane structure, but a *quantitative transformation* of the light incidence into an electrical equivalent of brightness. So, video positive feedback never overcomes the photosensor physical space. It is slightly different from the Larsen effect, because even when the microphony's feedback-signal achieves its maximum, limited by the equipment's capacity – electrical and digital –, the signal is not filling pre-established spaces, but it is *tensioning* the whole system: microphony will be maintained until the system collapses. On the other hand, even if video feedback has infinite recursion, it is always a recursive production of an image: it never saturates the pixels or video monitor.

In this sense, a photosensor does not enable the emergence of a complete membrane-like behavior. In fact, membrane porosity makes video's feedback sensitive to light incidence and to the interference of other objects: it is spatially sensitive. However, it does not form polarity due to the lack of tension that is only acquired in metastable equilibrium. That is, it is a pre-establish space to be filled, maintaining the feedback loop into a kind of discrete space-time, therefore confronting the premise of emergence by modeling approach: the allagmatic relationship.

6.3 Visible Ecosystems Modelization

Video feedback was set-up by positioning the digital camera in front of the video monitor, avoiding capturing its edge and then creating a monitor-inside-monitor pattern. In our spatially-extended system the digital environment needs to be approached in the limit of signal conversion. *Signal flow is more prevalent than the data it carries.* Video matrix is split in three planes – red, green and blue – creating a feedback signal for each plane due the camera-video feedback-loop. The hypothesis is to use the RGB color planes to digitally create a flow system that plays the role of membrane polarity, therefore allowing the generation of a metastable equilibrium. In the digital domain, each feedback signal operates in its neighbors positively and negatively at the same time it is

operated by them (Figure 3). This schema configures two sequences of signal inversion: the sequence R-, G+, B- in the upper operators, and R+, G-, B+ in the lower operators. Since in the loop every point can be the initial point, the order does not matter but the combination. This is crucial because the tension generated by these signal inversions is responsible for polarizing the membrane, as it will be shown below.

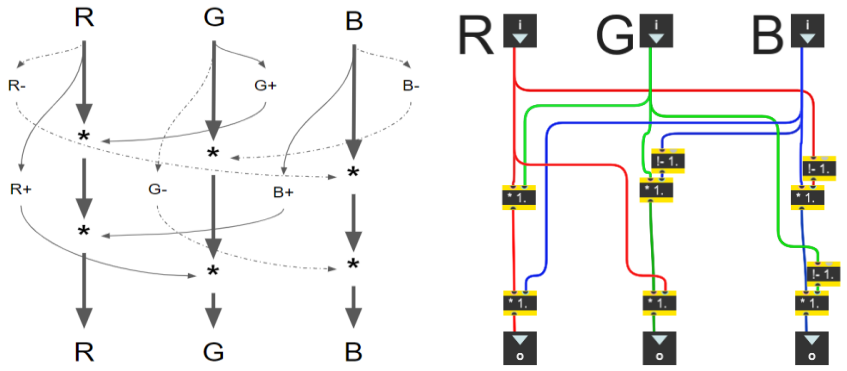


Figure 3. Visible ecosystem's digital mapping flowchart (left) and its implementation in MAX/MSP software (right). RGB signals are digitally mapped both directly (continuous-lines) and inversely (dashed-lines) connected by lower and upper operators of each signal creating two loop sequences, respectively, R+, G-, B+ and R-, G+, B-.

In a closed system this model would remain in stable equilibrium, but in an open system differences of light incidence during image capturing – the instabilities – provoke distinct changes in each color plane, resulting in a constant adaptation in contrast with the external conditions. There is no object between camera and video. The only information in the feedback-loop is the color combination and pixel frames. As the process of changing emerges, the gradient color nuances reveal the energetic flow: the membrane working inside a topologically defined space.

6.4 Experimental Evaluation

Considering the audible ecosystem model [Thomasi, 2022b], we assume the evaluation of the visible model from three features: a) the sensitivity to spatial organization; b) the emergence of feedback topologies; and c) the emergence of a metastable equilibrium condition. Since the video feedback itself creates spatial-sensitive dynamics – fulfilling the first two evaluation features – we establish as criteria for metastable condition the very emergence of the membrane-like behavior. It can be observed as a behavioral change in both spatial distribution and feedback topology. Thus, sensitivity to spatial organization was measured by changing light incidence straight to the video monitor with a cell phone flashlight and a ring light. The locality of light incidence over the video monitor surface is critical. Light becomes an attractor element inducing the organization of the RGB's signal flow. Depending on the conditions, the attractor can influence the evolution to a certain color gradient, provoking changes by disturbing the current state. Figure 4 shows snapshots of an experiment that moves the attractor point from the lower corner to the center of the monitor and turns back. As the ring light moves the color gradient

changes due the spatial sensitivity. However, when the attractor goes back to the corner, the gradient color becomes similar to the initial one when it was in the same position. It evidences a membrane-like behavior where the ring light is an external tensor: in contrast with a one-directional recursive distribution, there is a contrary force pushing back the signal flow. This experiment can also be seen in the Video 3.

Figure 4 - Snapshot sequence of a 40-seconds gesture illustrating the straight relationship between the ring-light movement – working as attractor point – and the RGB signal flow schematic that results in the color gradient. Light is projected straight to the video monitor. Source: the author.

The feedback topology formation (Figure 5) is another example of spatial sensitivity. In this experiment, a fixed attractor – a cell phone flashlight – and two digital cameras distinctly positioned were put in front of the video monitor. By switching the feedback-loop from one camera to another, the perspective of the attractor point and of the video monitor space is altered, and color nuances appear as result. Not only color gradient but the color distribution along the monitor is important. Video feedback in metastable equilibrium preserves certain properties that remain in the signal flow – i.e. information currently circulating through the feedback-loop and the tendencies toward stable states that the system has already shown. This can be interpreted as a kind of historicity, rather than complete changes as if nothing had happened before. Such response to the optical environment brings the visible ecosystem closer to the audible model already developed. By changing the feedback topology it is possible to achieve distinct color formations just as in the case of sound spectra.

Figure 5 - Snapshots from two cameras A and B illustrating two different feedback topologies. The fixed light point is visible in the upper middle region of the screen. Changes in the feedback topology are notable not only in the color sequence, but also in the distribuition of the color gradient: in the Camera A the brighter part is centered, while in the Camera B it tends to the left side. Source: the author.

7. Final Considerations

Artificial ecosystems based on spatially-extended systems appear as fruitful experimental territory for artistic research, offering other perspectives about dynamic systems studies, which include interactive systems. The proposed models are an open-ground for exploration of multimedia systems for artistic performance as well as a territory of convergence for experimental and theoretical debates in the area of ecologically-grounded system thinking. Rather than dualistic and parametric paradigms, these artificial ecosystems offer a feedback-based model that highlights the historicity along the structuring process with big potential to be applied in other fields of music and sound art. The implementation of the audible and visible ecosystem models have supported our structuring hypothesis based on metastable condition, membrane-like behavior and feedback topology. In this context, it was presented the emergence by modeling approach that furnishes a qualitative perspective of interactions and a schematic thinking to engender relationships. The problem of metastability in the visible ecosystem implementation appeared related to the notion of membrane polarity and it was solved by digitally implementing a self-regulating system based on the energetic-flow system: the spectral formations of the color gradient emerge from an operative schematics, as in the Simondon's Allagmatic theory. There are no pre-programmed directions or rules, but a simple engine of signals that operate on each other directly or through a structure – i.e. the membrane and the surfaces. These experiments go deeper in the notion of metastable equilibrium, bringing to the concrete plane critical debates often restricted to the theoretical sphere. We hope these findings may contribute to the improvement of interactive systems for artistic performance research.

8. Acknowledgements

We appreciate the support granted by a CNPq Fellowship, Project No. 150154/2022-4, and by the School of Communications and Arts at the University of Sro Paulo (ECA/USP).

References

Assis, P. "Virtual Works, Actual Things" In: Assis, P. (ed.). Virtual Works Actual Things: Essays in Music Ontology. Orpheus Institute Series. Leuven University Press, 2018.

Chembo, Y. "Optoelectronic Oscillators with Time-delayed feedback". Reviews of Modern Physics, 91. 2019.

Connors, T. "Audiovisual Installation as Ecological Performativity: A creative research practice". Doctoral thesis. University of Waikato, 2017.

Couchot, E. "Da representazro a simulazro: evoluzro das tйcnicas e das artes da figurazro". In: Imagem mбquina: a era das tecnologias do virtual. Org. Andrй Parente. Rio de Janeiro, Editora 34, 1993.

Crutchfield, J. “Space-time Dynamics in Video Feedback”. Physica D, 10, 1984.

Crutchfield, J. “Spatio-temporal complexity in Nonlinear image processing”. IEEE Transactions on Circuits and Systems, 35, 1988.

Di Scipio, A. “Sound is the interface: from interactive to ecosystemic signal processing”. Organised Sound 8, 2003.

Dupuy, J. “Nas origens das cıкncias cognitivas”. Trans. Roberto L. Ferreira. Sro Paulo: UNESP, 1996.

Eargle, J. The Microphone Book. Focal Press, Elsevier, 2005.

Filho, O. and Neto, H. Processamento Digital de Imagens. Brasport, 1999.

Fromm, J. “Types and forms of emergence”, In arXiv: Adaptation and Self-Organizing Systems, 2005 <arXiv:nlin/0506028v1>.

Hooker, C. “On the Import of Constraints in Complex Dynamical Systems”. Journal of Foundations of Science, 18 (4), 2012.

Ingold, T. “Estar Vivo: Ensaios sobre movimento, conhecimento e descrizro”. Trad. Fбbio Creder. Petrypolis: Editora Vozes, 2015.

Jones, S. “Organizing relations and emergence”. Artificial Life, 8, p. 418-422. MIT Press, 2002.

Keller, D. and Capasso, A. “New concepts and techniques in eco-composition”. Organised Sound 11, (2006).

Keller, D. and Lazzrini, V. “Ecologically Grounded Creative Practices in Ubiquitous Music”. Organised Sound 22(1), p. 61−72, 2017.

Kendall, G. “The Decorrelation of Audio Signals and Its Impact on Spatial Imagery”. Computer Music Journal, 19, (4), p. 71-87, 1995.

Lloyd, A. “The Coupled Logistic Map: A Simple Model for the Effects of Spatial Heterogeneity on Population Dynamics”. Journal of Theoretical Biology, 173, p. 217-230, Academic Press Limited, 1995.

Lidwell, W.,Holden, K., Butler, J. Universal Principles of Design. Rockport Publishers, 2010.

Prigogine, I. “Time, Structure and Fluctuations”. Science, 201, 1977.

Prigogine, I. “Exploring Complexity”. European Journal of Operational Research, 30, 1987.

Prigogine, I. “O fim das certezas: tempo, caos e as leis da natureza”. Trans. Roberto L. Ferreira. Sro Paulo: Editora UNESP, 2011.

Rand, D. “Measuring and characterizing spatial patterns, dynamics and chaos in spatially-extended dynamical systems and ecologies”. Phil. Trans. R. Soc. Lond. A, 348, p. 497-514, 1994.

Rheinberger, H. “Consistency from the perspective of an experimental systems approach to the sciences and their epistemic objects”. Manuscrito — Revista Internacional de Filosofia 34 (1), p. 307-321, Campinas, 2011.

Sauvanargues, A. “Crystals and Membranes: Individuation and Temporality”. Trans. Jon Roffe, in Gilbert Simondon: Being and Technology, ed. Arne de Boever et. al. Edinburgh University Press, 2012.

Schmidt, J. “Is there anything new under the sun? Instability as the core of emergence”. Emergence and Modularity in Life Sciences, Springer Nature Switzerland AG, 2019.

Simondon, G. “A individuazго a luz das noзxes de forma e de informazго”. Trans. Vera Ribeiro. Sго Paulo: Editora 34, 2020a.

Smalley, D. “The listening imagination: listening in the electroacoustic era”. Contemporary Music Review, 13, 1996.

Thomasi, R. “Estudos Ecos e vislumbres de uma performance musical ecossistкmica”. Tese de doutorado. Escola de Comunicaзxes e Artes, Universidade de Sго Paulo, Sго Paulo: USP, 2022b.

Thomasi, R.; Faria, R. R. A. “Patterns of emergence: Musical form over feedback”. In: Electroacoustic Music Studies Network, 2021, De Monfort, 2021c.

Thomasi, R.; Faria, R. R. A. “Sound feedback control model for live electronic performance in the Ecos Study”. REVISTA VЧRTEX, v. 9, p. 1-22, 2021b.

Thomasi, R.; Faria, R. R. A. “Moving Along Sound Spectra: An Experiment with Feedback Loop Topologies and Audible Ecosystems”. In: Proceedings of the International Computer Music Conference 2021, 2021a.

Vanag, V.K. “Study of spatially extended dynamical systems using probabilistic cellular automata”. Physics - Uspekhi 42 (5), p. 413 - 434, 1999.

Varela, F. “The Re-Enchantment of the Concrete: Some Biological Ingredients for a Nouvelle Cognitive Science”. In: Steels, L. and Brooks, R. The Artificial Life Route to Artificial Intelligence. Routledge, 1995.

Wilensky; U; Rand, W. “An introduction to agent-based modeling: Modeling natural, social and engineered complex systems with NetLogo”. MIT Press, 2015.

Young, Y. “Sound in structure: applying spectromorfological concepts”. Proceedings of the 2005 Electroacoustic Music Studies Network, 2005.

Video 1: Audible ecosystem’s dynamics viewed by sound spectra formations

Video 2: Visible ecosystem’s dynamics viewed by color gradient formations.

Video 3: Visible ecosystem with mobile attractor (ring light)

Non-human companionship: Practicing free improvisation through interaction with machines

Felippe Barros[1], **Sérgio Freire**[2], **Leandro Costalonga**[3]

[1]Programa de Pós Graduação em Artes - UFES

[2]Escola de Música - UFMG

[3]Departamento de Computação e Eletrônica - UFES

felippe.barros@edu.ufes.br, sfreire@musica.ufmg.br, leandro.costalonga@ufes.br

Abstract. *Free improvisation is a musical genre that defies traditional forms of music-making, defining itself by its absences. This creates a particular environment that imposes challenges to those trying to learn it and to educators inspired to teach it. Generally, ensemble practice is considered to be the most suitable and widely accepted method for practicing the genre. Nevertheless, should it be the only one? This article discusses the different approaches to learning improvisation and particularly free improvisation, proposing the use of interactive musical systems as a possible alternative to the problem. To do so, we present a custom-designed improvisation machine as a model example and a preliminary evaluation done with it.*

1. Introduction

After more than five decades of continuous practice, free improvisation still enjoys a curious position in the world of music: it's often misunderstood by other musicians, largely ignored by the general public, and deeply loved by a small sample of listeners and practitioners. Free improvisation can be best defined by its processes and intentions rather than its sonic output [Borgo 2022, 167], [Stenström 2009, 105] [Hickey 2009, 294], flourishing from the desire to be a universal form of music making, which borrows stones from each player's cultural background as its means to construct a new house for each performance. One can probably come up with some musical analysis to explain how a particular performance was built and which materials sustain it, although this analysis will probably not be useful when applied to future performances.

Unlike most music genres, which typically rely on fixed compositions, free improvisation is conceived as spontaneous creations, and differently from other types of music which do employ improvised material, the musical form and the development of a freely improvised piece is done in real-time, with no settled prior references (such as motifs, melodies, or chord progressions). Nevertheless, having no predetermined structure does not mean its practitioners have no preparation for public performance: most of them have found a way to establish themselves as improvisers, and as we will see further in the paper, this is generally thought to be achieved by repeated ensemble practice.

But how does a musician with no access to fellow improvisers can accomplish some degree of proficiency in this particular genre? This article exposes the first steps in a research that addresses this particular issue, proposing the use of improvising machines

as a possible solution. The study was conducted with the development of a prototype for an improvising machine, designed for this particular musical purpose, and evaluated in a preliminary, informal setting.

Improvising machines are a type of musical agent [Tatar and Pasquier 2019] designed for the task of live improvisation, typically in interaction with a musician. The development of such systems has been a recurring topic in computer music research, dating back to the pioneering work of George Lewis in the 80's [Lewis 2000]. The theme has spawned interest both in musicians and programmers, and has evolved to utilize various computational methods that reflect the musical and cultural backgrounds of their designers, ranging from complex rule-based systems to modern artificial neural network projects [Tatar and Pasquier 2019].

Engenhoca, a prototype for an improvising machine, is presented here for the first time as ongoing research that investigates a potential solution for individuals who wish to practice free improvisation alone. The project is distributed as free and open-source software (FOSS) and was originally initiated as an undergraduate thesis in music education. It is currently being further developed as part of a Master's research project. A technical description of the system and a preliminary evaluation will be presented, along with plans for its future development.

2. Pedagogical methods for musical improvisation

To begin our discussion on the position of improvisation in music education, we could first understand it as a musical skill analogous to spontaneous speech [Johnson-Laird 2002, 417]. This analogy, repeatedly employed by improvisers and researchers in the field [Dobbins 1980] [Burton 2010] [Berkowitz 2016], is defended by many as a justification for the necessity of learning to improvise, framing improvisation as a fundamental musical skill that is necessary for demonstrating creative ability in music. Under this perspective, a professional musician with proficiency in score reading who does not know how to improvise would appear as an adult who could read sophisticated poetry (and maybe write some too) while still not being able to sustain a dialogue [Dobbins 1980, 37].

The case for improvisation could also be sustained by the fact that most music performed and listened to is based on improvised genres, making a compelling case for, at least, the presence of improvisation in general musical training. The paradox appears when music education curricula do the opposite of what would be expected, with almost no space at all for improvisation courses in Western education [Sawyer 2007].

The development of pedagogical methods for improvisation has been analyzed by Pressing (1987) and summarized in five historical stages by Hickey (2009) : embellishment (1), patterns and models (2), problem-solving (3), play-by-ear (4), and free improvisation (5). In pre-Baroque times (1), improvisation was considered a "real-time composition" and was limited to variation and embellishment techniques. The use of patterns and models (2) from the 17th and 18th centuries, such as figured bass and Italian Partimento, find parallels with approaches based on imitation of melodic patterns from the 20th century. Problem-solving (3) exercises (pioneered by Jaques-Dalcroze) require technique and expressiveness of the students, with profound attention to their individuality. Methods focused on transcribing from recordings or teacher's performances (4) require students to infer which decisions to make in different musical situations. A humanistic point of view

is observed in modern approaches based on creativity and individual expression concepts (5), inspired by the work of Orff, Kodaly, Suzuki, Jaques-Dalcroze, and Schafer.

More recent approaches to music education have followed technological developments of the digital age, making use of a variety of available software to enhance the acquisition of a number of musical skills[1]. For instance, Fern (1995) proposes a computer program to teach jazz improvisation, using Miles Davis' 1954 recording of the tune *Four* as a transcribed reference for students and a chord-scale[2] method as a guide to creating new improvised lines. With the advent of the World Wide Web, some educators have turned to social media platforms such as YouTube as a means of sharing instructional material with a broader audience at little to no cost. Unsurprisingly, the platform has been used to teach musical improvisation, with a diverse range of methods employed, e.g. theory related to improvisation practices, performance transcriptions, technical exercises, and play-along videos, making it one of the most prominent ways in which the Web has contributed to the task. Finally, Frankel (2010) notices that even though musicians demonstrate pioneerism in the adoption of new technologies, there appears to be a continued reluctance among music educators to embrace technological tools for instruction.

3. Challenges to teach free improvisation

One might notice that most modern approaches to improvised music cited before belong specifically to the language of jazz, and that could be explained by its popularity among improvised musical genres, as well by the fact it consists of a musical idiom with approachable boundaries for harmonic, rhythmic and melodic discourses.

Returning to Johnson-Laird's analogy between musical improvisation and spontaneous speech, how does it fit free improvisation, a form of improvised discourse that does not belong to any specific idiom, does not adhere to any kind of syntactic rules nor uses any predetermined vocabulary? As Derek Bailey notes, it "has no stylistic or idiomatic commitment. It has no prescribed idiomatic sound. The characteristics of freely improvised music are established only by the sonic-musical identity of the person or persons playing it" [Bailey 1993, 83]. This paradigm doesn't find a parallel in spoken language, making evident the challenge to formulate methodologies for free improvisation education capable of attending to so many absences. This situation could explain why Borgo (2022, 9) presumes you would incur a passionate argument when asking an improviser if it can be taught. One way or another, the history of the genre has provided many examples of improvisers inspired to do so [Lange 2011].

A less idealist view on free improvisation "language" is offered by Menezes (2010, 15), when pointing out that practitioners of the genre do find their own boundaries shaped by their technical constraints and their personal database of experiences. And as even Borgo (2002, 184) indicates, there have been identifiable characteristics in the output of the genre in the last decades of its practice, with a shared range of "techniques, approaches and clichés" [Menezes 2010, 16], although not explicitly specified by the au-

[1] An extensive list of examples in various categories can be seen in Nart (2016). Fein (2017) presents an introduction to some of these tools, such as auto-accompaniment, notation, and music production software, explaining ways they could be used to complement jazz education in private, classroom, or lab settings.

[2] "Chord-scale" is a conventional designation for several methods in jazz education that associate chords found in a musical piece to specific musical scales to be employed in an improvisation.

thors. Thus, the absence of rigid characteristics in free improvisation does not necessarily imply a complete lack of boundaries.

Furthermore, one could argue the idiom of a particular performance and its set of rules is developed on the spot as an implicit agreement between the musicians, that "should be able to select the proper constraints in the course of the piece, rather than being dependent on precisely chosen ones" (Ann Farber apud [Belgrad 1999]). Under this perspective, a freely improvised piece should not employ prearranged material, but the decisions made by each player can shape the musical boundaries of the performance.

This context helps us to understand why free improvisation is seen to be dependent on interaction with other musicians, with collective engagement being accepted as the most fruitful context for it. For instance, Bailey presents that improvisation "although a vehicle for self-expression, is about playing with other people and some of the greatest opportunities provided by free improvisation are in the exploration of relationships between players" [Bailey 1993, 105]. In a complementary way, Tom Hall believes that when "more than one person is involved, improvisation is as much about the relationship between what's played as it is about what each person is playing" [Hall 2009].

Even though Bailey (1993, 105) asserts that the majority of improvisers explore the option of performing alone, Corbett (2016, 119) suggests that there is a certain skepticism regarding solo playing and raises the question of whether it might be more accurately viewed as real-time composition. Corbett's argument highlights the importance of interaction in free improvisation, implying that instead of interacting with other players, a solo improviser can engage with their instrument, the performance space, or the audience, a perspective that could be well integrated with Ubimus' technologies and practices.

To address those difficulties and particularities of the genre, approaches to the practice of free improvisation often emphasize the significance of ensemble practices. This approach draws parallels with language learning focused on conversation, emphasizing the importance of interaction to the genre. Stenström (2009, 38) defends the ensemble practice approach, arguing that solo playing may not provide the most solid foundation for ensemble improvisation, as the latter relies heavily on the musical interplay and interaction among its participants.

Although the argument for the importance of ensemble practice can be restrictive under a broader notion of creative models, free improvisation figures itself as a genre that, like Ubimus practices, should not limit musical diversity, which is exemplified by the increasing presence of musicians with varied backgrounds coming into the genre in the latest decades [Borgo 2022, 5]. Free improvisation also incorporates some of the same aspects of musical creation adopted by Ubimus[3], such as the focus on the creative *process* instead of the musical *product*, and an anti-formalist approach to music making that embraces and deals with the presence of technological and environmental resources and the participation of the audience.

Hall (2009) offers an alternative method to learn free improvisation, compiling exercises, and warm-ups that cover practices for groups, duets, and even solo playing. The book first presents his philosophy of improvisation and ideas for approaching its practice, which is later followed by dozens of exercises subdivided into thematic chapters. His

[3]See [Keller et al. 2014] for further discussion on the topic.

teaching seems to be preoccupied with the awareness needed for improvisers to accomplish fluid interactions and subtle explorations of the sound materials employed during a performance. The themes employed in each chapter explore distinct musical parameters or musical concepts such as melody and accompaniment, textures, groove, silence, and so on. Hall suggests, with examples, that the exercises can be combined to offer new possibilities for practice.

Methods used to teach free jazz can be valuable alternatives to learning free improvisation, even though both genres are not always considered as the same thing[4] [Corbett 2016, 17]. Crook (2006) offers a guide to learning free jazz (also referred to as free improvisation in the book) by departing from the understanding of conventional jazz. For him, one must be familiar with what he is *freeing himself from* to be truly *free*. Later on, the book demonstrates how improvisation can be organized when the elements of time and changes (musical pulse and chord progressions) are absent.

Vesisenaho et al. (2017) present a framework for the use of Information and Communication Technologies (ICT) for creativity related to improvisation. They argue that such technologies have the potential to introduce new possibilities for learning environments and their associated use with mobile networking could allow those experiences to take place in diverse educational settings. Under this premise, Clemente et al. (2020) conducted a series of experiments with ICT and Ubiquitous Computing in a gaming format to support free improvisation practices with an ensemble of saxophone, observing improvements in the sonority of the group, according to the students and teachers who participated in the experiments.

Barbara Lange (2011) noted that improvisers' educational values, as expressed in workshops she attended, align with Paulo Freire's educational principles that form the basis of Ubimus dialogics, which highlight "the role of the horizontal exchanges among group members, the respect for cultural diversity and the adoption of a positive attitude toward local knowledge" [Keller et al. 2020, 9]. However, Lange also observed that free improvisation workshops frequently revolve around a teacher figure who imparts information in a one-way direction, replicating the banking model that Freire criticized.

3.1. Machine improvisation as a valuable approach

For several decades, most people had not encountered any serious problems gathering together to practice most music genres. This scenario changed in 2020 due to measures taken in most countries to reduce the spread of the new coronavirus, with lockdowns having a brutal impact on people's ability to gather and play music together. Alternatives to in-person practice were explored during the period, with low-latency network communication being demonstrated as a somewhat promising solution for the task, even in contexts that demanded precise synchronous interaction [Bosi et al. 2021]. Playing remotely over the internet has been a practice in Ubimus even before the pandemic [Schiavoni et al. 2019], suggesting this can be a reasonable solution to bring together musicians to perform improvised music despite being physically separated. Software applications that enable remote communication or perhaps Internet forums dedicated to the practice could facilitate the assemblage of like-minded free improvisation enthusiasts interested in practicing from home.

[4] A distinction that has been criticized, see [Banerji 2021].

However, this approach still heavily depends upon live interaction, and as others have noticed, the inclination towards synchronicity may not be in Ubimus's practices best interest, especially in cases with "uneven levels of musical training" as pedagogical situations may impose [Keller et al. 2020, 4]. Thus, one may inquire if live interaction is needed for a freely improvised music session to occur. Dynamics that differ from synchronized performances could bring a new dimension to the experience, providing different ways to think about it and making room for new pedagogical concepts that may arise from such scenarios.

Additionally, it is remarkable that ensemble practice in nearly every genre of music is not to be considered the only source of training in a specific style. Most genres that function around a fixed composition employ repetition and memorization as part of training that should be performed alone. In contrast, when accounting for technical exercises, there is not much left for practitioners of free improvisation to explore while unaccompanied, and as we have seen above, solo playing does not provide much ground for the development of the interaction skills needed for the genre.

Researchers on interactive musical systems have long been interested in the potential applications of their innovations, ranging from education to different stages of creative practice. The underlying premise is that designing and engaging with such systems can yield valuable insights into our musicality [Rowe 2001, 3]. George Lewis, a leading figure in the field of improvising machines and a pioneer in research in this area, shares the conviction that these systems can help us understand the inner workings of music, and particularly of improvisation. For Lewis, the ability of machines to improvise provides us with a unique opportunity to gain insight into the mechanics of musical meaning [Lewis 2018, 5]. Lewis also argues that the act of developing such machines is a type of musical discourse that goes beyond the conventions of Western music practice [Lewis 1999, 108], questioning the model of musical interaction conceptualized on the passiveness of computer programs merely thought as instruments, in conformity with Ubimus concepts [Keller et al. 2020, 5].

After these considerations, we propose the use of improvising machines as a valuable approach to the practice of free improvisation without human companionship, believing it should serve as an alternative to ensemble playing, mimicking the companion of other improvisers or even providing a different kind of interaction than what is possible with human company.

4. Engenhoca

To address this idea, research has been conducted on the development of a prototype for an improvising machine, named *Engenhoca*, a Brazilian Portuguese jargon for a machine built in an improvised manner. The prototype is capable of generating music expected to be properly fitted in the context of free improvisation. *Engenhoca* is a rule-based[5] transformative system[6] that mirrors musical concepts and personal aesthetic preferences

[5]Rule-based refers to systems that execute predetermined actions in case some scenario or condition is met. The algorithms used in each system can vary greatly and are associated with the design preferences of its authors [Tatar and Pasquier 2019].

[6]Transformative systems refer to computer-based systems that manipulate existing musical material through various transformations to create new musical sentences, which may or may not be recognized as originated from the original material [Rowe 1992, 7].

of the author [Barros 2021]. The system analyzes incoming sounds played by a musician; decides what to play and whether or not to play; and finally synthesizes and outputs audio in real time. The system tries to derive its musicality from the musician it is playing along with, while also avoiding imitation or conformity, which provides it with a sense of agency and independence.

Engenhoca was entirely programmed in Pure Data (Pd)[7], a FOSS application, easily accessible by other musicians who might be interested in developing the system's core functionalities. To facilitate this, *Engenhoca* is also distributed as FOSS[8], allowing its code to be studied, executed, shared, and modified. We believe that this type of software distribution resonates with the principles of improvisation pedagogy, and with the very idea of what freedom is.

4.1. Technical description

The basic architecture of the system is subdivided into three main subpatches[9], responsible for different musical tasks: audio analysis (1), decision-making (2), and audio output (3). The separation between tasks makes it possible to implement a full range of audio synthesizers or MIDI instruments that can be controlled by the decision-making process.

The system settings can be controlled through a graphical interface, with controls designed to manage signal gain, calculate note offset amplitude (defining what is silence and what is sound), and select the type of output (digital audio synthesis or MIDI messages). The interface also offers simple oscilloscopes and volume meters for the audio input and the audio output.

Engenhoca works straightforwardly, analyzing the audio input with a diverse range of musical information retrieval algorithms, then segmenting and classifying them into musical phrases and types of sound events (1). This analysis is used by the system to make musical decisions (2) in two levels: an orchestration of the number of different voices playing, and each note's characteristics (duration, pitch, intensity, timbre, and octave), later on grouped into musical phrases and sent to virtual players (3), which can execute them through digital audio synthesis or MIDI messages.

The input analysis (1) subpatch is capable of discerning different characteristics of timbre, pitch, attack, rhythm, articulation, dynamics, and phrasing of incoming audio. Some higher-level algorithms were implemented directly in Pd, while others use Pd's native solutions or objects from the external library *timbreID*[10]. A representation of this subpatch can be seen below (see Figure 1).

The decision-making (2) subpatch is subdivided into two main processes: an orchestration subpatch that decides how many voices should be playing, and different subpatches for each voice (comprehending five independent monophonic virtual players). The decision on when to open and when to close a voice output is made by a series of calculations based on the musical activity of the incoming audio. The opening or closing

[7]Pure Data (Pd) is a visual programming language for multimedia designed by Miller Puckette. For more information, visit http://msp.ucsd.edu/ or https://puredata.info/.

[8]Available at `https://github.com/ofefo/engenhoca`.

[9]Applications developed with Pd are usually referred to as patches and organized into subpatches, in a logic that might resemble a cabinet with different drawers, each containing different functions or algorithms.

[10]For more information regarding this library, see [Brent 2010].

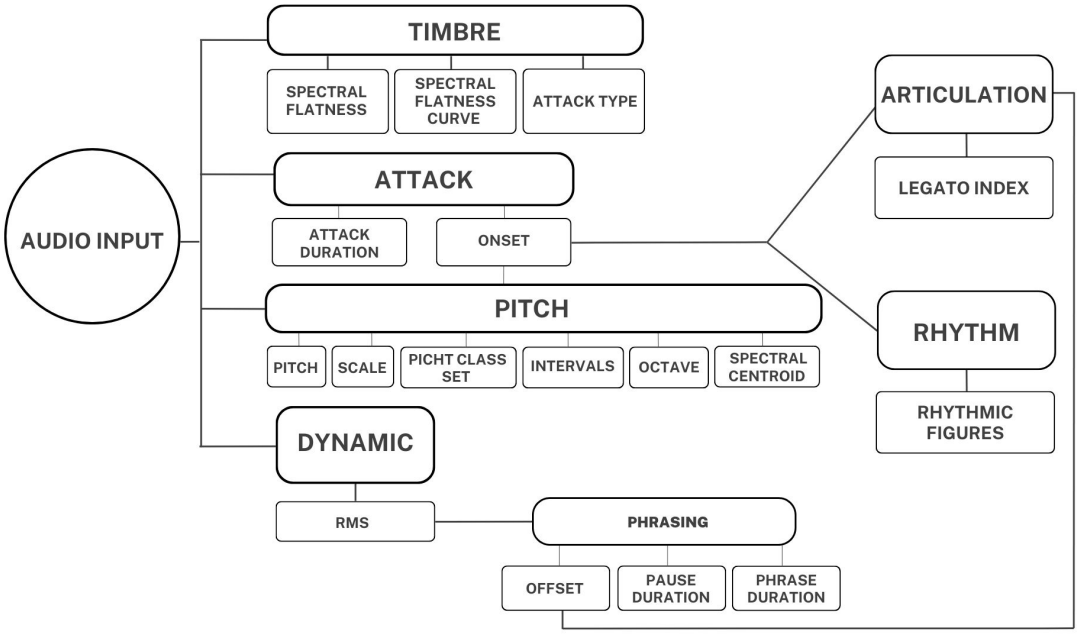


Figure 1. Dataflow representation of the audio analysis subpatch. The incoming audio is analyzed by different subpatches that measure the musical or sonic attributes of each phrase played by the musician it interacts with. The onset and offset of a musical event play an important role in defining when to listen to new attributes.

of a voice is allowed when intensity, intervallic relationships, or rhythmic data from what is being played changes drastically when compared to the last phrases heard by the system. The characteristics of each note that will be played are defined separately by each voice subpatch, selecting meaningful elements of musical input and performing varied transformations on it.

Rhythmic transformations are applied over a list of inter-onset intervals (IOI) extracted from the input, generating musical phrases with different sizes that vary in response to the average IOI duration of the last phrase played by the human. Examples of operations done are inversion, *ostinato*, rotation, shuffling, addition (in allusion to the added value rhythm process described in [Messiaen 1944, 16]), and multiplication (following Stockhausen's scale of duration [Stockhausen 1957], constrained by the duration range found in the incoming rhythm).

In a similar way, pitch transformations are applied over pitch sets that are extracted from the input. When no discernible pitch is identified in the input, the output also tends towards more percussive timbres, using the Spectral Centroid as a reference for pitch. The pitch operations are analogous to those described above for rhythm, but happen to be in a melodic context, in a style similar to what is commonly used when composing serial atonal music. Examples of operations are sorting, sorting followed by inversion, transposition, and multiplication. These same operations can be done to their complementary set [Forte 1973, 209], providing further harmonic contrast.

The output of the system (3) works in two different forms: a digital audio synthesizer and MIDI messages. The audio output consists of a Frequency Modulation

synthesizer as described by Chowning (1973) , routed through stereo spatialization and a Schroeder reverberator [Schroeder 1962], allowing a somewhat diverse sonic output. However, as we will address later, this has been shown to be aesthetically insufficient for extended use of the system. The MIDI messages provide the user an opportunity to use their own solution for sonic output. The audio input and the output are automatically recorded as raw files for each session.

4.2. Preliminary evaluation

The system was once evaluated in an informal manner, providing some modest feedback for the ongoing phase in which improvements are being made as part of a Master's research. We invited three undergraduate music students who had previous experience in free improvisation ensemble practice to play with the system for about a week, making recordings and observations of their experience. The evaluation was done in an interview format, with two open-ended questions about their experience with the system and perspectives about using similar software as a practice tool for free improvisation.

When asked about the system, problems recognized by the participants included the feeling their timbres were "repetitive" and their responses "not natural" or "random". One participant indicated that it was difficult to find an "endpoint" to the improvisation, and trying to "force it to react" in a desired way ultimately failed. When asked about the prospects of improvising machines as a practice tool, one of the students indicated that he finds it an "interesting resource", however, he "was not confident" about the aesthetical results of the interaction. Another participant had a "positive evaluation", but also thought the interaction to be more "difficult" than those with humans. The third participant saw "potential" for the idea, but felt the "lack of a corporeal presence" as a downside to the interaction[11].

The current evaluation of the system indicates that there is a general positivity towards the idea of improvising with a machine as a practice tool in free improvisation studies. However, since it was done with a small number of participants, informally interviewed, it is unclear whether its results show any real significance to the theme.

5. Discussion and future perspectives

As we have seen, pedagogy for improvised music genres has varied throughout music history, providing us with a range of possible approaches when we desire to teach them today. However, free improvisation presents some unique challenges to musicians and educators trying to work with it, as there's no correct way to approach it and it might not share an explicit musical vocabulary as found in other improvised genres.

To overcome these particularities, most improvisers accept ensemble practice as the most suitable method for practicing the genre. This research debates whether there's no alternative to ensemble practice in this context, observing different approaches for free improvisation education found in the literature, as well as suggesting technology-assisted practices such as network communications and the use of interactive musical systems as possible substitutes that conform with Ubimus' practices. Finally, we present *Engenhoca*, a custom-designed improvising machine under ongoing development. The project aims

[11]Audio demonstrations of interactions with the system can be accessed at: `https://archive.org/details/engenhoca`.

to demonstrate how such machines could be implemented, by distributing it as a FOSS application and permitting other musicians to modify the application for their particular musical goals and preferences.

As seen above, a noticeable problem in the interaction with the system lies in the lack of timbre variety. This was an expected opinion since the beginning of the development and can be explained by the simplicity of the audio synthesis methods used. As *sound* itself might be considered the basic unit in free improvisation [Hall 2009], a significant improvement to be made is to guarantee a broader sound scope for the system. Also, the perception of their responses being random might indicate the need for further improvements in the approach to interaction.

We expect to provide future insights into the pedagogical use of such systems by developing and evaluating them further. Moreover, to reach its goal of a viable pedagogical tool, we expect to provide *Engenhoca* with different functionalities that could help this ambition. Discussed ideas are statistical analysis of musical data that has been played and network communication connection that could enable different modes of interaction.

References

Bailey, D. (1993). *Improvisation: its nature and practice in music*. Da Capo Press.

Banerji, R. (2021). Whiteness as improvisation, nonwhiteness as machine. *Jazz and Culture*, 4(2):56–84.

Barros, F. (2021). *A improvisação de máquina como suporte à prática da improvisação livre*. Undergraduate thesis, Universidade Federal de Minas Gerais.

Belgrad, D. (1999). *The Culture of Spontaneity: Improvisation and the Arts in Postwar America*. University of Chicago Press.

Berkowitz, A. (2016). The cognitive neuroscience of improvisation. *The Oxford Handbook of Critical Improvisation Studies*, 1:56–73.

Borgo, D. (2002). Negotiating freedom: Values and practices in contemporary improvised music. *Black Music Research Journal*, 22(2):165–188.

Borgo, D. (2022). *Sync Or Swarm: Improvising Music in a Complex Age*. Bloomsbury Academic, 2 edition.

Bosi, M., Servetti, A., Chafe, C., and Rottondi, C. E. M. (2021). Experiencing remote classical music performance over long distance: A jacktrip concert between two continents during the pandemic. *Journal of the Audio Engineering Society*, 69(12):934–945.

Brent, W. (2010). A timbre analysis and classification toolkit for pure data. In *Proceedings of the 2010 International Computer Music Conference, ICMC 2010, New York, USA, 2010*. Michigan Publishing.

Burton, G. (2010). Gary burton: Improv class. `https://www.youtube.com/watch?v=t2txO_u2eNg`.

Chowning, J. M. (1973). The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, 31(7).

Clemente, M., Falleiros, M., Tavares, T., and Fornari, J. (2020). Experiments on technology-assisted free improvisational practices with an ensemble of saxophone. In *X Ubiquitous Music Workshop*. Zenodo.

Corbett, J. (2016). *A Listener's Guide to Free Improvisation*. University of Chicago Press.

Crook, H. (2006). *Beyond Time and Changes: A Musician's Guide to Free Jazz Improvisation*. Kendor Publishing.

Dobbins, B. (1980). Improvisation: An essential element of musical proficiency. *Music Educators Journal*, 66(5):36–41.

Fein, M. (2017). *Teaching music improvisation with technology*. Oxford University Press.

Fern, J. L. (1995). The effectiveness of a computer-based courseware program for teaching jazz improvisation.

Forte, A. (1973). *The Structure of Atonal Music*. Yale University Press.

Frankel, J. (2010). Music education technology. *Critical Issues in Music Education: Contemporary Theory and Practice*, pages 236–258.

Hall, T. (2009). *Free Improvisation: A Practical Guide*. Bee Boy Press.

Hickey, M. (2009). Can improvisation be 'taught'?: A call for free improvisation in our schools. *International Journal of Music Education*, 27(4):285–299.

Johnson-Laird, P. N. (2002). How jazz musicians improvise. *Music Perception: An Interdisciplinary Journal*, 19(3):415–442.

Keller, D., Costalonga, L., and Messina, M. (2020). Ubiquitous music making in covid-19 times. In *10th Workshop on Ubiquitous Music (UbiMus 2020)*.

Keller, D., Lazzarini, V., and Pimenta, M. S. (2014). Ubimus through the lens of creativity theories. In *Ubiquitous music*, pages 3–23. Springer.

Lange, B. R. (2011). Teaching the ethics of free improvisation. *Critical Studies in Improvisation/Études critiques en improvisation*, 7(2).

Lewis, G. E. (1999). Interacting with latter-day musical automata. *Contemporary Music Review*, 18(3):99–112.

Lewis, G. E. (2000). Too many notes: Computers, complexity and culture in "voyager". *Leonardo Music Journal*, 10:33–39.

Lewis, G. E. (2018). Why do we want our computers to improvise? In *The Oxford Handbook of Algorithmic Music*. Oxford Handbooks.

Menezes, J. (2010). *Creative process in free improvisation*. Master's thesis, University of Sheffield.

Messiaen, O. (1944). *The Technique of my Musical Language*. Alphonse Leduc.

Nart, S. (2016). Music software in the technology integrated music education. *Turkish Online Journal of Educational Technology-TOJET*, 15(2):78–84.

Pressing, J. (1987). Improvisation: Methods and models. In Sloboda, J. A., editor, *Generative Processes in Music*, pages 129–178. Oxford University Press.

Rowe, R. (1992). *Interactive Music Systems*. MIT Press.

Rowe, R. (2001). *Machine musicianship*. MIT Press.

Sawyer, K. (2007). Improvisation and teaching. *Critical Studies in Improvisation/Études critiques en improvisation*, 3(2).

Schiavoni, F., de Faria, P. H., and Manzolli, J. (2019). Interaction and collaboration in computer music using computer networks: An ubimus perspective. *Journal of New Music Research*, 48:1–15.

Schroeder, M. R. (1962). Natural-sounding artificial reverberation. *Journal of the Audio Engineering Society*, 10(3):219–223.

Stenström, H. (2009). *Free ensemble improvisation*. Tese de doutorado, Academy of Music and Drama; Högskolan för scen och musik.

Stockhausen, K. (1957). How time passes... *Die Reihe*, pages 10, 40.

Tatar, K. and Pasquier, P. (2019). Musical agents: A typology and state of the art towards musical metacreation. *Journal of New Music Research*, 48(1):56–105.

Vesisenaho, M., Dillon, P., Sari, H.-N., Nousiainen, T., Valtonen, T., and Ruolan, W. (2017). Creative improvisations with information and communication technology to support learning: A conceptual and developmental framework. *Journal of Teacher Education and Educators*, 6(3):229–250.

Paper Session

Ubimus Creative Practices, Influences and Tools

Atmospheric Attunement
weather data sonification with ubiquitous sensor nodes

Juan C. Duarte Regino

Aalto University
Department of Art and Media
P.O. Box 11000 (Otakaari 1B)
FI-00076 AALTO

Abstract. *This paper presents a case of artistic research focused on listening to atmospheric processes, particularly reviews an installation called Augury. This work merges contemporary digital sound techniques and weather sensor computing with ancient weather divination methods to create an experience of atmospheric attunement. The installation underscores the use of sensor nodes in the exhibition space's vicinity, enabling real-time registration of weather events. The project also explores atmospheric processes, by bridging modern technology with ancestral wisdom to enhance our understanding of a more-than-human environmental perspective. Atmospheric attunement refers here to our ability to synchronise with and gain familiarity with our surroundings, to foster a deeper empathy with our environment through technologically mediated and embodied listening.*

1. Introduction

This article delves into the design of an auditory display interface inspired by ancient weather divination practices and artifacts. It provides in-depth insights into sonification and spatialization techniques that enable immersion in atmospheric processes. Previous articles related to this project have covered its artistic background [Duarte Regino 2023d] and design overview both in previous [Duarte Regino 2023a] and post [Duarte Regino 2023c] exhibition stages. This article focuses on the iterative design process, with a special emphasis on interactive sonification and ubiquitous computing sensor networks.

Within the context of *Augury*, I employ a semantic approach to sonification [Walker and Nees 2011], translating weather data into audible signals that establish a meaningful connection with indirect ecological and metaphorical environmental sound categories. Additionally, I present the mapping between weather data and the sound parameters utilized in the sonification process.

Augury aims to enhance our perception of atmospheric processes, which can be subdivided into notions such as attunement, resonance, and harmony. In this article, I specifically explore the concept of attunement from the perspective of ubiquitous computing, drawing on Mark B. Hansen's proposal of ubiquitous computing as a medium that enhances our attunement to the environment through a model of micro-temporality [Hansen 2013].

2. Artistic research background of Augury

Augury embarks on an exploration of weather divination practices and artifacts originating from ancient meteorology. This practice, belonging to ornithomancy, involved the observation of birds in Ancient Roman society, influencing decisions ranging from everyday activities like farming to state-level matters such as urban planning and warfare [Marcattili et al. 2020]. The parapegmata served as an inscription device made of stone to record astral events believed to impact the weather, enabling the identification of patterns across different cities within the ancient Roman Empire [Taub 2004].

In the Aztec culture, the Smoking Mirror (Tezcatlipoca) symbolized the winds and the subconscious, among other agencies. Like other scrying practices, it revolved around the observation of obsidian mirrors, allowing individuals to not only see their reflection but also perceive the world beyond human capacity [Ackermann and Devoy 2012].

These historical practices inspired my artistic research, leading me to integrate ubiquitous weather sensing systems and sonification techniques. In the case of Augury, I employed custom-made sensing stations symbolizing bird-emissaries, conveying weather insights from distant locations. The parapegmata was re-imagined as a surface for holding stones, facilitating interaction with weather-related events. Similarly, the Smoking Mirror served as a central metaphor to unify the experience of listening to the atmosphere, by enabling to trigger the installation and becoming immersed into a media experience with sound, light and smoke [Duarte Regino 2023a].

2.1. Interfacing media with Ancient Meteorology

In this research, interfacing entails bridging ancient knowledge with digital technologies to explore alternative ways of understanding our connection to the environment, particularly the atmospheric order. Throughout history, this order has been examined through methods, tools, and rituals for divination, all speculating on its intricate relationships and dynamics [Taub 2004].

Within this investigation, the aim is to restore these weather divinatory narratives and make them accessible in the present, allowing for physical and interactive experiences through multisensory technologies. Viewing these narratives as unique methods for designing media experiences within an ecological framework, this approach also promotes the incorporation of culturally situated knowledge [Haraway 2013], such as indigenous perspectives [Robinson 2020], to diversify our current technological landscape [Hui 2021].

3. Atmospheric attunement with sensing technologies

This artistic research introduces atmospheric attunements as a focal point for exploring weather-related indicators, particularly those with short to midterm impacts and temporalities. In contrast to climate studies, which involve a long-term and intricate temporal framework spanning across time and space. Attunement refers here to our ability to synchronise with and gain familiarity with our surroundings. This research hypothesis holds that this kind of experience with the atmosphere can foster a deeper empathy with our environment.

To create atmospheric attunement experiences, this research explores modern methods of sensing our atmosphere through ubiquitous computing systems, while high-

lighting their divergence from human-dependent weather sensing and prediction approaches. Contemporary meteorology has widely adopted remote sensing as a standardized practice, primarily driven by the need for environmental computing and the monitoring of atmospheric processes [Gabrys 2016]. This shift eliminates human subjectivity from weather sensing and prediction, enabling sensor-computing models to collect unbiased information.

Notably, sensing technologies operate within significantly smaller time scales than humans, prioritizing sensing over perception and consciousness [Hansen 2013]. Furthermore, networked sensor technologies can seamlessly coordinate their efforts across extensive territories, providing a distributed capacity to detect phenomena larger in scale than humans, such as atmospheric processes.

3.1. Ubiquitous computing of the environment

Within this context, ubiquitous computing takes form as sensors, receptors, actuators, and even cognitive processes, establishing a comprehensive framework for atmospheric sensing across vast geographical areas [Hayles 2017]. Similar to the approach in ”Augury,” the interest in sensing weather is supported by technical systems operating within the dynamic and ever-changing atmosphere, seamlessly blending ”natural” environments with synthetic cognition.

Mark B. N. Hansen’s description of ubiquitous computing systems [Hansen 2013] emphasizes their micro-temporality in sensing and catalyzing sensations, in contrast to human processes of consciousness and perception. Ubicomps utilize non-conscious cognitive processes to establish a direct and immediate connection with the sensory world, synthesizing information peripherally without the need for conscious representation and memory, as humans do. Consequently, ubiquitous systems enable the catalysis of sensation at a finer temporal granularity beyond human capacity. By leveraging and synthesizing information as peripheral, these systems facilitate new experiential cognition of environmental processes, including atmospheric phenomena central to this artistic research.

Katherine Hayles’ concept of non-conscious cognition is essential for understanding the mechanisms behind the synthetic sensing systems proposed in this research to attune to our atmosphere [Hayles 2017]. It recognizes emergent effects, fluid mutations, and transformations. Although these technical systems remain beyond human perception, they imbue our atmosphere with meaning in technical terms, relying on somatic markers such as chemical or electrical signals that align with their systems. This allows computational media to adapt to a changing environment like the atmosphere.

Furthermore, I believe that Hansen’s emphasized microtemporality and the capability to sense across vast geographical areas are critical elements for fostering atmospheric attunement through the real-time sonification of weather events. Beyond the technical mediation of sensing our atmosphere, this research also explores embodied ways of attunement [Förster 2021], particularly focusing on affective relations and becoming familiar with the atmospheric milieu. While this aspect exceeds the scope of the present article, I aim to establish a coherent connection between mediated and embodied modes of attuning to our atmosphere in future publications.

4. Interaction design system

Augury is structured as a multidimensional system encompassing remote sensing stations, real time sonification processes, touch-interaction, interactive lighting, and smoke-mist elements. The initial project iteration utilized weather data collected from four locations in the vicinity of the exhibition space, specifically at the RIXC gallery in Riga, Latvia, during the first week of May 2023. In this part I will describe how each component works and its holistic relation. There is video documentation of the project online, which gives an impression of the experience [Duarte Regino 2023b] [Duarte Regino 2023e].

4.1. Environmental drone semantics

Since there is no direct sonic link between the narratives of ancient meteorology and the project's suggested sound design. The approach to follow is focused on creating a sonic environment that connects to both deep-time meteorology and contemporary climate change awareness. Drone music the aesthetic chosen for this project, is in words of the musicologist Joanna Demmers, "a straight line of sound that marks the edge between the present and future, presence and absence, essential and incidental disruptive"[Demmers 2015]. It also resonates with the ritualistic tone underscored by the divinatory practices that grounds this project.

Augury relies on weather data that either has been recently recorded (for prototyping) and real time data (for its final deployment). Hence, it contrasts with an approach to climate data sonification [Vogt and Visda 2013]. Nevertheless, it attempts to bring a listening experience about weather phenomena on a more-than-human scale through an earned micro-temporality that is mediated by ubiquitous sensor nodes. In this first version of the sonification deployment, employed semantically a kind of indirect metaphorical, in the case of drone music sounds, and indirect ecological approach, in the case of the noise bursts. Akin to this approach, in the field there has been Ecologically Grounded Creative Practices in Ubiquitous Music [Keller and Lazzarini 2017]. Wherein, sonic ecologies encompass material resources, human agents and the context where the creative activity takes place [Keller et al. 2014]. I look into *Augury* as a meeting of a tangible interface, triggered by listeners, and as context the surrounding weather data becoming audible.

In *Augury*, various weather elements are artistically transformed into sound through a sonic style influenced by drone music. This includes eolian harp-like sounds and noise bursts that evoke atmospheric phenomena like lightning and strong winds. This project draws inspiration from minimalist compositions by Eliane Radigue [Radigue et al. 2009], as well as soundscape works by Hildegard Westerkamp [Westerkamp 2002], shaping the atmospheric sound design.

4.2. Sensing stations

In the initial iteration, data collection involved wind direction readings from four points, temperature, barometric pressure, and dust particle detection. These sensors were connected to an Arduino Nano through a custom-made circuit. The data was recorded using the OSC protocol for transmission to a laptop running Pure Data. To establish a time reference, a basic timestamp was added to indicate the recording's start time. Over five days, approximately four hours of weather data were sampled. In total, an amount of 5 different datasets with over 20 hours were recorded for the initial setup of *Augury*.

Wind direction was determined by electret microphone sensors positioned to face cardinal points, with their readings subsequently averaged to identify the prevailing wind direction. While this method may capture unwanted acoustic signals from the surroundings as wind force, future iterations will involve comparing this measurement with other types of wind sensors to validate the desired signal capture.

The next version of the sensing stations will feature a communication protocol system based on LoRa (Long Range) radio communication across few kilometers to operate through a mesh network that share data across four sensing nodes [Foubert and Mitton 2020]. This new feature expects to fulfill the need for a reliable real-time system of weather stations covering a meso-scale of atmospheric processes.

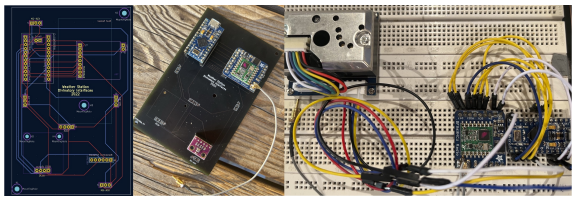


Figure 1. Prototypes of sensor circuits

4.3. Sonification

Previous related works have explored sonification of weather events through sensor registers [Flowers et al. 2001] and employed tangible interfaces in their design [R Ness et al. 2010]. In this project, the sonification process was implemented in Pure Data and the ELSE library, to read data recorded offline, and enabling to acquire a set of synthetic instruments which align with the preferred drone music aesthetics. Changes in the wind direction were translated into discreet parameters using a waveform based granular synthesizer called grain.synth . The raw measure across four points were mapped to predetermined pitch values and duration of the synthesizer grains. Each of the different days of recorded data used a set of different waveforms to be played by the granular synthesizer and produce and distinctive quality of sound to contrast each of the days. Similarly other projects with a profile for ecological sound design have employed granular synthesis as a tool for composition [Keller and Truax 1998] and [Keller 2000].

Barometric pressure, along with data on dust particles, was transformed into dynamic noise bursts resembling atmospheric events. To achieve this, a Lorenz Chaotic Generator with adjustable Hz rate acted as a noise source. It's worth noting that Edward Lorenz made significant contributions to the study of mathematical models related to the atmosphere, discovering the set of strange-attractor differential equations, which is relevant to this project's context [McWilliams 2019]. Following the chaotic generator, a reverb unit with controllable decay duration was influenced by the dust particles, mapped to a scale of 0 to 1 values. The resulting sounds from this part of the sound modules occasionally resembled thunderstorms and at other times conveyed a moderate wind turbulence.

The slight temperature variations registered in the data were re-sampled into spatialization parameters across four speakers. Thus, temperature values were scaled from 0 to 1, where in the bottom value signified an equal distribution of four different sounds across speakers, and 1 would mean an all mixed distribution.

4.4. Touch-interaction

These databases ran continuously in a loop, but users had the ability to select which dataset they wanted to listen to through a touch interaction facilitated by obsidian pieces and a mirror. This touch-based interaction was achieved by attaching copper strips to the obsidian pieces, which were connected to a touch-sensor module. The touch-sensor module, in turn, was connected to an ESP32 micro-controller, which transmitted OSC messages indicating which obsidian piece the user was touching.

In future iterations of the project, there are plans to incorporate multiple touch points per obsidian piece, allowing for more granular control over each dataset. Additionally, the touch interaction is envisioned to encourage a slower, more deliberate interaction, in contrast to instantaneous responses. This approach underscores the ritualistic nature of the artwork, emphasizing the importance of attentive perception of atmospheric events.

Figure 2. Touch interaction with obsidian pieces *Augury*

4.5. Smoke and Light Interaction

The smoke interaction is central in this project, to evoke the divinatory practice known as the smoking mirror. It creates a misty ambiance, complemented by interactive lights, forming a magical circle within the exhibition space where the interaction unfolds. Whenever the system detected a user touch, it had the potential to trigger bursts from a DMX-controlled smoke machine located at the base of the installation pedestal. Additionally, an array of LED addressable tubes suspended from the ceiling formed a light installation capable of generating various light compositions within the exhibition room. These compositions were customized to correspond to specific days, varying in color, intensity, and the duration of flickering events. While the initial prototype lacked a direct connection between the datasets and light parameters, future iterations are anticipated to establish a meaningful link between them.

5. Conclusions

In this paper, I have presented an overview of the design of *Augury*, a project that synthesizes concepts from ancient and contemporary perspectives on weather sensing and divination, with the aim of creating an atmospheric attunement experience. The examination of Ubicomps by Hansen supports the exploration of technical systems' potential in mediating attunement to atmospheric processes, leveraging the attributes of microtemporality and pervasiveness.

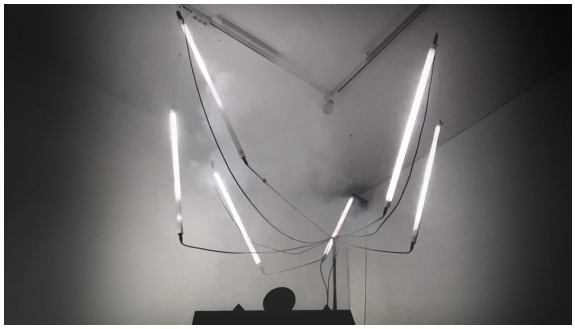


Figure 3. Smoke and Light Interaction in *Augury*

Following the acquisition of weather data, the sonification process contributes to providing a semantic understanding of each dataset, drawing inspiration from the aesthetics of drone music and ecological soundscapes. In this regard, this research contributes elements to the field of Ubiquitous Music, particularly within the realm of Ecologically Grounded Creative Practices. This article also signifies a milestone in shaping the direction of the upcoming design iteration of the project.

6. Acknowledgements

Augury was produced at the RIXC Centre for New Media Culture with received funding for residential visits from the Nordic-Baltic Mobility Programme for Culture to establish project "RIXC Art Science Residencies". Thanks to Anton Filatov for the designing the light interaction.

References

Ackermann, S. and Devoy, L. (2012). 'the lord of the smoking mirror': Objects associated with john dee in the british museum. *Studies in History and Philosophy of Science Part A*, 43(3):539–549.

Demmers, J. (2015). *Drone and Apocalypse: an exhibit catalog for the end of the world.* John Hunt Publishing.

Duarte Regino, J. C. (2023a). Augury : an interface for generating soundscapes inspired by ancient divination. In *NIME 2023 Conference Proceedings*.

Duarte Regino, J. C. (2023b). Augury: Hybrid listening atmospheric attunement. Available at `https://www.youtube.com/watch?v=SGafXzLUwSc&feature=youtu.be`.

Duarte Regino, J. C. (2023c). Computing atmospheric attunement and hybrid listening through augury and scrying. In *School of XCOAX*.

Duarte Regino, J. C. (2023d). A hybrid listening to atmospheric processes. In *ISEA 2023 Conference Proceedings*.

Duarte Regino, J. C. (2023e). Smoking mirror. Available at `https://www.youtube.com/watch?v=yfC_AehYAXU`.

Flowers, J. H., Whitwer, L. E., Grafel, D. C., and Kotan, C. A. (2001). Sonification of daily weather records: Issues of perception, attention and memory in design choices. *Faculty Publications, Department of Psychology*, page 432.

Förster, D. (2021). *Aesthetic experience of metabolic processes*. meson press.

Foubert, B. and Mitton, N. (2020). Long-range wireless radio technologies: A survey. *Future internet*, 12(1):13.

Gabrys, J. (2016). *Program earth: Environmental sensing technology and the making of a computational planet*, volume 49. U of Minnesota Press.

Hansen, M. B. (2013). Ubiquitous sensation: Toward an atmospheric, collective, and microtemporal model of media. *Throughout: Art and culture emerging with ubiquitous computing*, pages 63–88.

Haraway, D. (2013). Situated knowledges: The science question in feminism and the privilege of partial perspective. *Women, science, and technology*, 3:455–472.

Hayles, N. K. (2017). *Unthought: The power of the cognitive nonconscious*. University of Chicago Press.

Hui, Y. (2021). *Art and cosmotechnics*. U of Minnesota Press.

Keller, D. (2000). Compositional processes from an ecological perspective. *Leonardo Music Journal*, 10(1):55–60.

Keller, D. and Lazzarini, V. (2017). Ecologically grounded creative practices in ubiquitous music. *Organised Sound*, 22(1):61–72.

Keller, D., Lazzarini, V., and Pimenta, M. S. (2014). Ubimus through the lens of creativity theories. In *Ubiquitous music*, pages 3–23. Springer.

Keller, D. and Truax, B. (1998). Ecologically-based granular synthesis. In *ICMC*.

Marcattili, F. et al. (2020). Driediger-murphy, lindsay gayle (2019). roman republican augury. freedom and control. oxford: Oxford university press. *ARYS. Antigüedad: Religiones y Sociedades*, (18):400–405.

McWilliams, J. C. (2019). A perspective on the legacy of edward lorenz. *Earth and Space Science*, 6(3):336–350.

R Ness, S., Reimer, P., Krell, N., Odowichuck, G., Schloss, W. A., and Tzanetakis, G. (2010). Sonophenology: a tangible interface for sonification of geo-spatial phenological data at multiple time-scales. Georgia Institute of Technology.

Radigue, E., Fernandez, A., and Rose, J. (2009). The mysterious power of the infinitesimal. *Leonardo Music Journal*, 19(1):47–49.

Robinson, D. (2020). *Hungry listening: Resonant theory for indigenous sound studies*. U of Minnesota Press.

Taub, L. (2004). *Ancient meteorology*. Routledge.

Vogt, K. and Visda, G. (2013). Sonification of climate data. In *EGU General Assembly Conference Abstracts*, page 13482.

Walker, B. N. and Nees, M. A. (2011). Theory of sonification. *The sonification handbook*, 1:9–39.

Westerkamp, H. (2002). Linking soundscape composition and acoustic ecology. *Organised Sound*, 7(1):51–56.

AOGscript – Design of a Stand-Alone Scripting Language for the Generation of Music

Guido Kramann

Technische Hochschule Brandenburg

kramann@th-brandenburg.de

Abstract. *Through AOGscript, an extended version of AOG (Arithmetic Operation Grammar) is provided in this work. Unlike AOG, AOGscript now also offers the possibility to formulate repetitions and temporal dilations. This brings this script language a little closer to an ideal in which the symbolic representation of a script completely describes a musical composition and thus acquires a representational character. This opens up new possibilities for representing and passing on music, it improves reactive work in live performances, and it guides the user to new ideas by recognizing the variation possibilities in script parts he knows and anticipating their musical result.*

1. Introduction

In the context of ubiquitous music, special importance is attached to comprovisation, i.e., a practice in which a composition emerges only in the course of a performance as a joint product of the actors involved. In this process, the actors can be together on location [Clemente et al. 2020], or they can be in places far away from each other and interact with each other via the Internet [Frazier-Reed 2021] [Messina et al. 2021]. The people actively involved can be professional musicians or programmers [Stolfi 2020] [M. Messina 2020], or amateurs [Kramann 2021]. In any case, this practice requires as preliminary work the preparation of the musical material then used. The material is not provided in the classical way in the form of sheet music, but typically it is software that represents the possibility space of the performance, so to speak, and forms the core of the joint interaction in real time. Experience has shown that the following dilemma typically arises in the provision of this piece of software: either the musical material represented by the software in an abstract manner is very simple, easy to use, but at the same time trivial and aesthetically almost unusable, or conversely it is interesting, aesthetically appealing, but hardly masterable in the context of a live performance and/or for laypersons.

In this article, the scripting language AOGscript is proposed as a means to resolve this dilemma to some extent: If one first accepts that this scripting language is limited to the description of music with discrete pitch levels, it opens up to one the possibility of formulating also the generation of temporally very extended musical structures in a very compact way. In the form of the symbolic representation of the music to be generated, it can form a useful communicative basis for planning the course of a musical performance. In composition, in turn, it complements immediate musical feedback by providing a clear, compact, and thus easy-to-remember symbol-based visualization.

2. Motivation for the Development of a Generative Scripting Language for Compositions and Musical Performances

It is one thing to write generative software for creating music, but quite another to present the ideas it contains in a way that is comprehensible to third parties (Figure 1a). For third parties, such a program is typically a black box. To adress this issue parameters from the software that essentially determine the generated musical shape could be extracted (Figure 1b). At least this way the result can be influenced effectively and a user can learn something about the meaning of these parameters for the shape of the generated music via this feedback. Providing access to parameters via a sensor-based human-machine interface could be interpreted as an implementation of this principle [Rodrigues et al. 2017] [Desnoyers-Stewart et al. 2018] [Eaton and Miranda 2016].

Going further in this direction, parts of the software can be influenced by script parts by implementing so called modding. In this case the structure of the scripts can then already represent a part of the generative principles underlying the software (Figure 1c). Modding has existed since the start of computer music (e.g. in form of the ”unit generator” in MUSIC III [Mathews 1961]) and got very famous in game development like for Minecraft [overwolf 2021].

Finally, the original software can take a back seat to the script parts and the script becomes an independent language that can be described to third parties independently of the interpreter behind it (Figure 1d). Even though in csound, Pure Data (PD), MaxMSP, and Faust the source code is typically compiled and not interpreted, they detach from the underlying programming layer in the way meant here. These languages provide specific language elements suitable for music production, and they serve as the basis for many projects in computational music, and more specifically, in ubiquitous music [Lazzarini et al. 2014] [Anache 2017], [Michon et al. 2021].

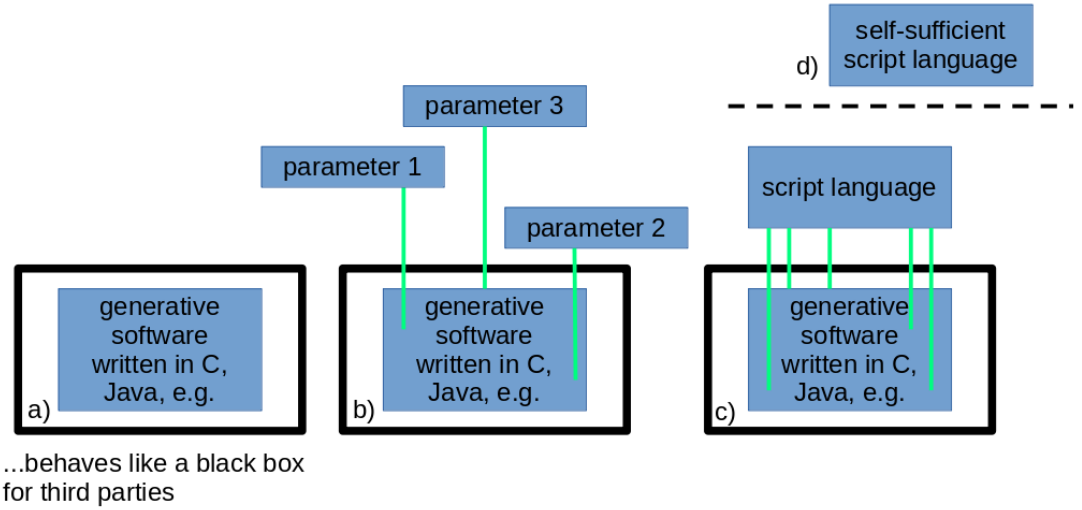


Figure 1. Making a scripting language independent.

If one comes up to this point, then this has the enormous advantage that the script

has representation character in relation to the generated piece of music. Knowing the script is sufficient to be able to reproduce the piece of music. The language elements of this script are then very powerful, or what is the same, slight variations of the linguistic expressions that can be formed with them result in significant law-like changes of the musical shape. In other words, the linguistic expressions are very sensitive to the musical form. This is accompanied by a further advantage: The user can quickly learn to implement musical ideas with the help of the quick successive additions and changes to such a script with continuous automatic conversion to music. This also makes the system suitable for live performances.

Compared to the previously mentioned languages csound, Pure Data (PD), MaxMSP, and Faust, which are very much focused on the formulation of the properties of sonic events, the script language AOGscript introduced in this article is mainly suited to generatively describe the temporal organization of entire compositions based on already available virtual musical instruments with their specific sonorities and discrete tone levels. This concentration on a limited task area, makes it possible to assign simple symbols to the available functions. This, in turn, allows extremely compact and thus quickly modifiable representations and their interpretation in real time. It should be noted that the speed at which a music production generator can be modified in an effective manner (in this case, via a script) and immediate sonic feedback is recognized as a crucial criterion for a user to advance to a level of proficiency at which he or she can confidently and intelligently use the tool.

With the help of AOGscript's predecessor, the script language AOG, it is possible to generate complete pieces of music to the greatest possible extent without the script having to be particularly long [Kramann 2020]. Ten to twenty lines are often sufficient. However, continued work with the language elements described so far has shown that there are still important aspects of music to which we as humans are accustomed, but which can be represented with AOG only with very great effort. Previously, these were typically added directly to the underlying software layer. This practice was found to be unsatisfactory, as it meant that not all of the essential aspects that were responsible for the resulting musical composition could be manipulated by the script.

This paper therefore attempts to extend AOG in such a way that, in the best case, supplementary components of hidden, directly programmed generative software can finally be dispensed with altogether. This is largely achieved by adding language elements that can now be used to describe repetitions and temporal dilations. Furthermore, a second script section is added at the back, in which functions can be specified, to which the individual results of the evaluated first script section can be assigned as transfer parameters. Certainly, the development of derivatives and variants based on AOG cannot be considered complete even with AOGscript. However, the current state of development opens up a wide field of composition possibilities. The following introduction to AOGscript should give a larger community the opportunity to explore and discuss these possibilities. This is done with a certain claim to completeness and independence, as expressed in (Figure 1d). A concluding discussion attempts to place the approach described here within a more general context of techniques and practices, all of which are intended to serve the purpose of teaching musical knowledge and compositional skills.

3. Description of the AOGscript Scripting Language

Code that generates music can be seen as concretizing a particular conception of what can be considered music. This conceptual notion is grounded in AOG in the fact that the distribution of small prime factors (2,3,5,7) in the sequence of natural numbers is already considered to be organized in a musically meaningful way, and original musical compositions can be obtained by transformations of this sequence [Kramann 2020]. With the script language AOGscript, interpreted line by line from top to bottom, such transformations can be described very easily. AOGscript is divided into two distinct sections that immediately follow each other. The first part forms the actual core of the language and the individual lines are composed of chains of operations, which are always applied to the previously obtained result. **The close relation to counting, i.e. to the temporal sequence of natural numbers, is taken into account in the first part of AOGscript by the symbol ' (apostrophe, decimal ASCII-code 39). For each loop pass in equidistant time steps (ticks) a counter is incremented and everywhere where the apostrophe appears in the script, it is replaced by this counter value. The script in turn is always interpreted completely at each tick and each line of the first part returns an integer greater than or equal to zero.** The first part corresponds largely to the ideal of an independent script language. The second part, on the other hand, is used to pass on the result values determined in each of the lines of the first part to a sound generation system. Each line of the second part therefore represents a function with certain passing parameters. On the one hand, constant values can be specified as parameters. As a second possibility, references to the results of the calculations of the first part can be specified. The rules according to which the first section is formed are the following:

- AOGscript is line oriented.
- The lines are evaluated in order from top to bottom.
- A complete evaluation takes place cyclically with each temporal tick.
- A tick counter starts at zero and increments with each tick.
- The current tick counter value can be specified in the script at any position by the character **'** (apostrophe, ASCII code 39).
- Each line has a positive integer or zero as result.
- The result of a preceding line can be accessed in a following line with [*line number*] (line number with counting from one in square brackets, ASCII codes 91 and 93).
- There are unary and binary operations.
- Operand (argument) of operations can be a constant integer greater than or equal to zero, the result of a preceding line, the current tick (accessible via **'**), or an expression in round brackets.
- Zero, one, or more, unary operations can follow an operand. The first is applied to the operand. The following ones are applied to the result of the preceding operation.
- To the left and the right of a binary operator there must always an evaluable sequence.
- Expressions in round brackets are evaluated first, otherwise there is no prioritization of operators, but the evaluation is always from left to right.
- Results less than zero are set to zero. For example, the unary signum operation $ for -1 and 0 returns 0 in both cases.

3.1. Unary operations in the first section at AOGscript

The four unary operations of AOGscript are described in table 1.

Table 1. Enumeration and Description of All Unary Operations of AOGscript

Character	ASCII Code	Description	Effect On -1,0,1,2,3,4,5,5,7,8,9,10,11,0
$	36	Signum.	0,0,1,1,1,1,1,1,1,1,1,1,1,0
_	95	Reduction. Maintains from a number only its powers of the primefactors 2,3,5, and 7.	0,0,0,2,3,4,5,5,7,8,9,10,0,0
!	33	Maintains the previous result, if the current is zero.	0,0,1,2,3,4,5,5,7,8,9,10,11,11
?	63	Sets the current value to zero if it is equal to the previous one.	0,0,1,2,3,4,5,0,7,8,9,10,11,0

3.2. Binary Operations in the First Section at AOGscript

There are currently 19 binary operators available in the current version of AOGscript. The first 4 correspond to the basic arithmetic operations and are shown in table 2. Another 12 operators are described in table 3. Finally, there are 3 novel binary operations that can affect the progress of a sequence of numbers over time, as each occurs when a line is repeatedly evaluated in AOGscript with each successive tick, see Table 4 . In the expression $xoperationy$, x represents the result of a sequence that has already been evaluated, which is to the left of the operation in question. y is to the right of the operation, and is then the parameter that changes the timing of x over several ticks. For example, if the operation is the character °, y represents the phase shift of the sequence of x in ticks. For example, if y is 3, the operation is not followed by the currently computed x, but by the one from 3 ticks ago. In order to enable the interpreter of AOGscript to implement the three operations that change the time behavior, it was necessary to record the progression with advancing ticks of the preceding sequence x in each case.

Table 2. Enumeration and Description of All Binary Arithmetic Operations of AOGscript

Character	ASCII Code	Description	Result from x *Operation* y
+	43	Addition.	x+y
−	45	Subtraction.	x-y
*	42	Multiplication.	x*y
/	47	Division of whole numbers. (Decimal places are truncated)	x/y

3.3. The second part of AOGscript representing functions

The second part each example of the type AOGscript consists of establishes a connection between the generated results of the first part and a sound generation system. The lines

Table 3. Enumeration and Description of All Non-Arithmetic Binary Operations of AOGscript.

Character	ASCII Code	Description	Result from x *Operation* y
:	58	Integer division. (is only executed if x is divisible by y without remainder)	x/y, if x modulo y = 0, else 0
\	92	Checks divisibility.	x, if x modulo y = 0, else 0
%	37	Modulo.	x modulo y
=	61	Checks equality.	x, if x=y, else 0
<	60	Checks, if $x < y$.	x, if $x < y$, else 0
>	62	Checks, if $x > y$.	x, if $x > y$, else 0
&	38	Minimum.	min(x,y)
|	124	Maximum.	max(x,y)
,	44	Powers of y in x. ($x = reminder * y^n$)	y^n, if $n > 0$, else 0
;	59	Exponent n of y^n in x. ($x = reminder * y^n$)	n, if $n > 0$, else 0
.	46	Removes all potencies of y in x. ($x = reminder * y^n$)	reminder, if $n > 0$, else 0
~	126	Constructs decimal digits d_i of x as as digits of a number in base y.	$\prod_i (d_i mody) * y^i$

Table 4. Enumeration and Description of All Binary Operations of AOGscript that Affect the Timing

Character	ASCII Code	Description	{0,1,2,3,4,5,6,7,8,9,10,11,...} *operation* 3
°	176	Phase shift.	{0,0,0,0,1,2,3,4,5,6,7,8,...}
@	64	Dilation.	{0,0,0,1,1,1,2,2,2,3,3,3,...}
§	167	Repetition.	{0,1,2,0,1,2,3,4,5,3,4,5,...}

of the second section in AOGscript immediately follow those of the first section. In the second section of AOGscript, each line consists of exactly one sequence of arguments separated by commas and enclosed in curly braces. An argument can be a positive integer greater than or equal to zero, or the evaluation result of a line of the first section can be accessed. The latter is done in the same way as for the first section by specifying the line number in square brackets. The lines of the second section correspond to special functions, which essentially control a midi instrument. The assignment of a line to a specific function is based on the number of parameters contained in the respective curly brackets. Whether it comes to the call of the corresponding function at the time of a certain tick depends on whether certain passing parameters designated as *sensitiv* are greater than zero at a certain tick. The various expressions in the second section of AOGscript, are deliberately given very specific meanings in this article. This is necessary so that a script as a whole can actually represent a piece of music. Specifically, the functions with one to four parameters in the meaning described here are sufficient to control a piano, see table 5. By addition of further functions the possibility is opened to use instead of it,

or additionally melody instruments, see table 6. In general, however, the assignment of concrete functions in the second part can be regarded as a freely designable element.

Table 5. Function Types of the Second Section of AOGscript in the Context of a Piano Control

Expression	Effect	Sensitive Parameters	Meaning of Parameters
$\{p\}$	sets beats per minute	p	p=beats per minute
$\{p, q\}$	sets a pedal at the piano	Function reacts if new value is not equal to the old one	p=pedal number p=0: sustain p=1: sostenuto p=2: soft p=3: harmonic q=value
$\{p, q, r\}$	converts (frequency*factor/100) to the nearest miditone and plays it. (Tone stops when new tone comes in same voice, or when maximum length reached.)	q	p=voice q=frequency r=factor
$\{p, q, r, s\}$	As before. In addition, there is a transposition by s semitones.	q	p=voice q=frequency r=factor s=transposition

3.4. Supplementary information about AOGscript

- Text after the # character and the character itself is interpreted as a comment.
- AOGscript does not use variables.
- AOGscript is fault-tolerant in that the evaluation result of a line is always set to zero if the evaluation fails.
- In a pre-evaluation, all invisible characters except line breaks are removed.
- To facilitate live coding, lines can be marked with a letter as the first character. The pre-evaluation removes these, but enters the corresponding line number for each subsequent line where the corresponding letter appears in square brackets.

3.5. Simple Control of a Piano with AOGscript

- The expressions of the second part of AOGscript with one to four parameters are suitable for controlling a Disklavier or a virtual piano.
- If a note of the same voice emerges, a lingering preceding note of the same voice is terminated, if necessary.
- With each appearance of a note of the same pitch at the same tick, the velocity or volume of the one note played by the piano increases.

3.6. Adding Melody Instruments

- The method of adding individual melody instruments works in the current version in a very specific, but also very systematic way: The piano notes to be played for a particular tick are pre-marked.

Table 6. Enumeration and description of all callable functions in the second section of AOGscript.

Expression	Effect	Sensitive Parameters	Meaning of Parameters
$\{p, q, r, s, t\}$	Configure sound selection for an instrument.	r	p=rank q=channel r=volume s=midi min t=midi max
$\{p, q, r, s, t, u\}$	Just like $\{p, q, r, s\}$. Additionally, there is an indication of the instrument's range.	q	p=voice q=frequency r=factor s=transposition t=midi min u=midi max
$\{p, q, r, s, t, u, v\}$	Like before. Additionally, there is an indication of volume.	q,v	p=voice q=frequency r=factor s=transposition t=midi min u=midi max v=velocity
$\{p, q, r, s, t, u, v, w\}$	Like before. Additionally, there is an indication of duration.	q,v	p=voice q=frequency r=factor s=transposition t=midi min u=midi max v=velocity w=duration

- Each occurrence of an expression with five parameters $\{p, q, r, s, t\}$ in the second part of AOGscript, adds another melody instrument, compare table 6.
- An active melody instrument then selects from this set of tones the one with the pitch closest to a tone played previously by that instrument.
- If it is the very first tone of this instrument, the distance to the mean between the lowest and highest playable tone of this instrument is taken as a basis ($s = midimin, t = midimax$).
- In case of two equal distances, the lower note is taken.
- On the piano, this note is then removed, respectively not played.
- If several additional melody instruments are active, the rank of each instrument can be specified in AOGscript ($p = rank$).
- The lower the rank of an instrument, the earlier it may select its note before the other instruments.

4. Implementation of AOGscript

In the spirit of ubiquitous music, the implementation was done with the goal of making the use of AOGscript accessible to as many people as possible. The goal was to find a good compromise between two conflicting objectives: On the one hand, there should be as few obstacles as possible to the use of AOGscript. Such hurdles can be an unintuitive or complicated user interface, but also demanding requirements for basic software or libraries to be implemented. On the other hand the implementation as such should be made open, in order to make also own developments possible by their analysis. This works better, however, if elaborate user interfaces are dispensed with and as much of the software as possible is provided by libraries already available from third-party suppliers. Finally, the sound of the produced music pieces should be appealing enough to motivate to use AOGscript.

An extreme simplification in the final implementation could be achieved by simply writing the script to be interpreted into a simple text file named code.txt. This is reloaded and interpreted by the software before each tick. Written code thus takes effect as soon as the text file is saved.

Java/Processing (version 3.5.4, see `https://processing.org`) was used as the development platform. To enable sound generation here, one version (AOGscriptMIDI) works with a midi library that has to be installed additionally (TheMidibus). A second version (AOGscriptCFE) makes use of a library for direct sound generation (ComposingForEveryone). The source code (processing sketches) of both versions can be downloaded from the author's homepage: `www.kramann.info/AOGscriptMIDI.zip`, resp. `www.kramann.info/AOGscriptCFE.zip`. These two links were provided mainly to give the possibility to analyze the interpreter of AOGscript and thus to allow own developments. In addition, AOGscriptCFE was exported for Linux and Windows as immediately executable software, see `www.kramann.info/AOGscriptCFELNX.zip` and `www.kramann.info/AOGscriptCFEWIN.zip`. These two variants are mainly intended to make it easy to get started with AOGscript. All four variants are provided under the MIT license, see for example `https://pitt.libguides.com/openlicensing/MIT`.

Note regarding musical performances: The music pieces generated by AOGscript do not have a natural end, since they are based on the sequence of natural numbers (the ticks) running towards infinity. However, switching mechanisms can be formulated with the AOGscript language that make the music stop at a certain tick.

5. An Introduction to AOGscript by Means of Small Consecutive Examples

The examples are intended to provide a step-by-step basic understanding of how AOGscript works. They are each accompanied by a short description and an audio file. The audio files were created with AOGscriptMIDI (see above) by controlling virtual instruments based on physical modeling.

Example Listing 1 is so to speak the Hello World example to AOGscript: The tick '= $\{0, 1, 2, 3, 4, 5, ...\}$ is reduced with _. This means only powers of 2,3,5 and 7 remain. The result of this reduction becomes a divisor of 2520. 2520 itself consists only of powers of 2,3,5 and 7, because 2520 = 2*2*2*3*3*5*7. Thus harmon-

ically related frequencies result, as long as the divisor is smaller than 2520, or otherwise zero. The sounding sequence of tones most likely represents something like the sound of the sequence of natural numbers. Here, line 1 produces from the sequence of ticks {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,...} the sequence of numbers {0,0,1260,840,630,504,420,360,315,280,252,0,210,0,180,168,157,0,140,...}. The numbers in this sequence that are greater than zero are then converted to midi tones in the last line and played. A sonic implementation with visualization can be found here: `https://youtu.be/e81wd1b3FEE`, audio file see `www.kramann.info/b1.mp3`.

Listing 1. Sonification of the Natural Numbers

```
2520/('_)    # Produce sequence of numbers (see text).
{0,1}        # The sustain pedal is set.
{72}         # The tempo is set at 72 beats per minute.
{0,[1],100} # Convert numbers from line 1 to midi tones and play.
```

Example Listing 2 extends the previous example in that now the number that determines which frequencies are available to play is slowly changed. Originally this was the constant 2520. Also, the basic sequence is transformed by applying the operations %12 and *3, audio file see `www.kramann.info/b2.mp3`.

Listing 2. Introduction of Transformations

```
('/3%6)+('/60) # siehe Text.
2520*[1]+9000/('_%12*3)
{0,1}
{72}
{0,[2],100}
```

Example Listing 3 adds polyphony and repetition structures, audio file see `www.kramann.info/b3.mp3`.

Listing 3. Introduction of Repetition Structures

```
('/3%6)+('/60)
2520*[1]+9000/('_%12*3) §4 §8
2520*[1]+9000/('_%16*3) §4 §12
2520*[1]+9000/('_%18*3) §4 §16
2520*[1]+9000/('_%24*3) §4 §24
{0,1}
{72}
{0,[2],100}
{1,[3],100}
{2,[4],100}
{3,[5],100}
```

Finally, example Listing 4 introduces four melody instruments and makes use of the possibility to mark lines with letters for easier handling instead of specifying them directly. In addition, the rank of the melody instruments is also controlled by the script, audio file see `www.kramann.info/b4.mp3`.

Listing 4. Addition of Melody Instruments

```
n ('/3%6)+('/60)
a 2520*[n]+9000/('_%12*3) §4 §8
b 2520*[n]+9000/('_%16*3) §4 §12
c 2520*[n]+9000/('_%18*3) §4 §16
d 2520*[n]+9000/('_%24*3) §4 §24
e 2520*[n]+9000/('_%24*3) §4 §32
f 2520*[n]+9000/('_%24*3) §4 §36
g [a]@2?
h [b]@3?
i [c]@6?
j [d]@8?
k [e]@12?
l [f]@16?
m '/24%6+5*3
{0,1}
{72}
{0,[g],[m],0,36,96}
{1,[h],[m],0,36,96}
{2,[i],[m],0,36,96}
{3,[j],[m],0,36,96}
{4,[k],[m],0,36,96}
{5,[l],[m],0,36,96}
{0,0,0,55,84} # rank,midichannel,volume,midimin,midimax
{1,1,0,50,79} # rank,midichannel,volume,midimin,midimax
{2,2,0,50,79} # rank,midichannel,volume,midimin,midimax
{3,3,0,50,72} # rank,midichannel,volume,midimin,midimax
```

6. Conclusions

Expressions formulated in any language always include the possibility of reaching what is meant. Taking up Edmund Husserl's idea of chains of reference, one could say that, starting from a symbolic script, an interpretation richly grounded in reality is constituted [Husserl 2009] across a chain of reference structures. Conversely, this interpretation must be able to withstand repeated comparison with its symbolic representation. Music is classically written down with notes. This notation can be interpreted as music, and this interpretation must in turn stand up to comparison with its symbolic representation. Many years of studying music enable people to contextualize notes, to derive semantic connections from them, and to give them a nuanced sound with the help of a musical instrument. People who study composition, in turn, classically learn theories and practices about how to provide music-representing notes.

In ubiquitous music, on the other hand, there is a long-standing practice of incorporating the skills necessary for composing and making music into tools, the use of which then opens up the possibility of producing music to non-professionals. The tools are typically created in languages such as Csound, Pure Data, or Faust [Lazzarini et al. 2014]. The resulting tool, or virtual musical instrument, can then be used in live performances, or can serve as the basis of amateurs for their everyday "little c" compositional practice

[Keller et al. 2014]. Thus, a tool, or virtual musical instrument in the sense described above, can be designed in such a way that almost any way of handling it will always pro-duce something that calling it music will elicit approval from a large number of people. To understand what is meant here, one can also consider the acoustic predecessors of such musical instruments, such as the Orff instruments, pandrums, calimbas and the like. With these instruments, the guarantee that ”beauty” will come out when you use them is bought by a severe limitation of playing possibilities.

The use of electronics and software now opens up the possibility of producing (partly virtual) musical instruments that behave similarly ”benignly” in handling, but which no longer, or at least to a lesser extent, rely on a restriction of playing possibilities to achieve this. This happens there, however, at the price that their behavior no longer depends in a simple and unambiguous way on their handling, but is, for example, adaptive, complementary and/or intelligent [Camporez et al. 2020]. The immediate sonic feedback of continued use, in such a case, often suffices to develop an intuitive understanding of adequate handling of such a musical instrument. In contrast to an incorporation of physical peculiarities and handling techniques in acoustic instruments, a (pre-symbolic) incorporation of music theory and compositional ideas takes place to a greater extent in virtual instruments [Magnusson 2009].

If such a musical instrument is made available in the form of a DIY project, then it may be possible for the future user to acquire at least the theoretical knowledge for its adequate handling by means of the intensive examination of the inner nature of the musical instrument that is necessary in a DIY project. But what is the relation between a pre-symbolic incorporation of music theory and compositional ideas and the learning of symbol-based algorithmic generative methods for music production, as exemplified by the script language AOGscript? For this purpose, it is proposed to apply what is stated at the beginning of the article about the relation between algorithm and script also to the relation of script and/or algorithm to the adaptive, complementary and/or intelligent musical instrument:

If it is possible to formulate adaptive, supplementary and/or intelligent behavior as an algorithm or, at a higher level of abstraction, as lines of code in a script language, then the user has been given a language that can show him the possible actions with the instrument immediately at a stroke. Why is that? – Well, because the formulated algorithm, and to an even greater extent the present special script, points beyond itself to all the variants, and alternatives, that would also be possible in its place. While a musical instrument in front of one is only what it is, algorithms and scripts are not only symbolic representations of a special aspect of the instrument, or of a composition, but here someone has already brought the special expression into a larger context by the use of the language alone and thus conveys a multiplicity of possibilities of action, which empowers the user to operate comprovisation himself and independently, or to develop adaptive, complementary and/or intelligent musical instruments himself. This assumption is supported to a certain extent by the thesis of cumulative learning, which has already been experimentally proven in the symbol grounding theory. Afterwards a new situation, in which a symbol appears, can serve to open an alternative also valid however so far unknown network of meaning connections and to find resulting from it a solution for a so far not yet mastered problem:

> ” *In real life, symbol grounding has a dynamic or contextual aspect to it. Heidegger refers to this as as-ness of language. In other words, language enables us to see the world (or the context) in a new way. Suppose Alice says to Bob, Í need a hammer.́ Bob, seeing no hammers around, hands her a rock. This is clearly a successful case of communication, even though the word ́hammerẃas grounded to a rock by Bob.* ”
> ([Swarup et al. 2006],p.188)

This empowerment of the user is taken to the extreme by a purely symbol-based scripting language like AOGscript: While in languages like Csound, Pure Data, or Faust, the representation of music is on a variety of levels, such as

> ” *words, menus, cables or icons on the one hand, or abstract representations such as variables, parameters and functions on the other* ”
> ([Magnusson 2009],p.174)

, in AOGscript it lies almost exclusively at the level of the one script itself. One might call this the WYSIWYH principle – **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **H**ear: whatever you want to accomplish sonically or in musical structure, you know from the start that it must be done at the symbolic level immediately in front of you. This promises to simplify the preparation phase of providing a setting for musical performance, but also to provide the necessary explanations to non-experts to introduce them to a compositional technique.

However, the endeavor to lift everything relevant for the description of the generation of a piece of music onto the symbol level always goes hand in hand with a certain limitation of the creative possibilities and also with a certain implicit normative definition of what counts as music. Here, one has to decide individually whether the advantages outweigh the disadvantages in the particular application.

7. Further Work

The division of AOGscript into two parts should be seen as an intermediate solution and a preliminary compromise on the way to eventually really representing all aspects of a musical structure in a generative script. In particular, the selection of certain tones from a pool of precalculated values would be a candidate for another operation in the first part of the script. However, its implementation would in principle require the processing of vectorial data with the operations as well. But this would affect the basic structure of the interpreter, which is why it would have to be completely rewritten. Apart from that, Processing is relatively easy to install, but presents a greater hurdle than, for example, a web application based on Javascript. So it stands to reason to work equally towards making the language more readily available. And finally, it makes sense to generate not only the sonic result but also the notation corresponding to a script, in order to establish a link between these two representation systems for music. In short, AOGscript is work in progress and because this is so, this progress will be made visible in the future on the following page: `www.kramann.info/42_AOGscript`.

References

Anache, D. (2017). Using pure data for real-time granular synthesis control through leap motion. In Aramaki, M., Kronland-Martinet, R., and Ystad, S., editors, *CMMR 2016, LNCS 10525 - Bridging People and Sound*, pages 171–179. Springer, Heidelberg.

Camporez, H., Silva, J., Costalonga, L., and Rocha, H. (2020). Robomus: Robotic musicians synchronization. Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020). url: https://hal.archives-ouvertes.fr/hal-02997201/document.

Clemente, M., Falleiros, M., Tavares, T., and Fornari, J. (2020). Experiments on technology-assisted free improvisational practices with an ensemble of saxophones. Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020). url: https://hal.archives-ouvertes.fr/hal-02997201/document.

Desnoyers-Stewart, J., Gerhard, D., and Smith, M. L. (2018). Augmented virtuality with a synchronized dynamic musical instrument: A user evaluation of a mixed reality midi keyboard. In Aramaki, M., Davis, M. E. P., Kronland-Martinet, R., and Ystad, S., editors, *CMMR 2017, LNCS 11265 - Music Technology with Swing*, pages 540–557. Springer, Heidelberg.

Eaton, J. and Miranda, R. (2016). The hybrid brain computer music interface - integrating brainwave detection methods for extended control in musical performance systems. In Kronland-Martinet, R., Aramaki, M., and Ystad, S., editors, *CMMR 2015, LNCS 9617 - Music, Mind, and Embodiment*, pages 132–145. Springer, Heidelberg.

Frazier-Reed, T. (2021). bandy - an interactive performance system. Proceedings of the 11th Workshop on Ubiquitous Music (UbiMus 2021).

Husserl, E. (2009). Logische untersuchungen teil2 - zur phänomenologie der erkenntnisstufen. pages 596–631. Meiner, Hamburg.

Keller, D., Lazzarini, V., and Pimenta, M. (2014). Ubiquitous music. pages 29–30. Springer, Heidelberg.

Kramann, G. (2020). Composing by laypeople: A broader perspective provided by arithmetic operation grammar. *Computer Music Journal*, 44(1):17–34.

Kramann, G. (2021). Kaleidophone - a collaborative web-based tool for comprovization based on arithmetic operation grammar. Proceedings of the 11th Workshop on Ubiquitous Music (UbiMus 2021).

Lazzarini, V., Keller, D., Pimenta, M., and Timoney, J. (2014). Ubiquitous music ecosystems: Faust programs in csound. In Keller, D., Lazzarini, V., and Pimenta, M., editors, *Ubiquitous Music*, pages 129–150. Springer, Heidelberg.

M. Messina, C. M. G. M. (2020). Contracapas for remote double-bass and effects: Creative semantic anchoring, corpus linguistics and remote interaction. Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020). url: https://hal.archives-ouvertes.fr/hal-02997201/document.

Magnusson, T. (2009). Of epistemic tools: musical instruments as cognitive extensions. In *Organized Sound 14(2)*, pages 168–176. Cambridge University Press.

Mathews, M. (1961). An acoustical compiler for musical and psychological stimuli. Bell Telephone System Technical Journal. url: https://worldradiohistory.com/Archive-Bell-System-Technical-Journal/60s/Bell-System-Technical-Journal-1961-3-Complete.pdf.

Messina, M., Aliel, L., Mejia, C. M. G., Filho, M. C., and de Melo, M. T. S. (2021). Live coding on orca, the geopolitics of the english language and the limits of creative semantic anchoring: A preliminary hypothesis. Proceedings of the 11th Workshop on Ubiquitous Music (UbiMus 2021). url:

Michon, R., Orlary, Y., Letz, S., Fober, D., and Dumitrascu, C. (2021). Mobile music with the faust programming language. In Kronland-Martinet, R., Ystad, S., and Aramaki, M., editors, *CMMR 2019, LNCS 12631 - Perception, Representation, Image, Sound, Music*, pages 307–318. Springer, Heidelberg. overwolf (2021). Curseforge - all mods. url: https://www.curseforge.com/minecraft/mc-mods.

Rodrigues, A. M., Zuffo, M. K., da Rosa Belloc, O., and Faria, R. R. A. (2017). A virtual musical instrument for 3d performance with short gesture: Exploring mapping strategies with virtual reality. In Aramaki, M., Kronland-Martinet, R., and Ystad, S., editors, *CMMR 2016, LNCS 10525 - Bridging People and Sound*, pages 301–315. Springer, Heidelberg.

Stolfi, A. (2020). Playsound agora. Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020). url: https://hal.archives-ouvertes.fr/hal-02997201/document.

Swarup, S., Lakkaraju, K., Ray, S. R., and Gasser, L. (2006). Symbol grounding through cumulative learning. In Vogt, P., Sugita, Y., Tuci, E., and Nehaniv, C., editors, *Symbol Grounding and Beyond*, pages 180–191. Springer, Heidelberg.

Auralization of Room Acoustics using Binaural Impulse Response

Joshua Daniel.M.C

Department of Music - National University of Ireland, Maynooth
Maynooth, Ireland

danyjosh68@gmail.com

Abstract. *This project aims to develop a software for auralizing an acoustic space using binaural impulse responses. In this case, the Octophonic studio at Maynooth University . The objective is to capture the room's binaural impulse response with a dummy-head microphone and simulate it through headphones by developing a software using Csound and Cabbage. The software applies the impulse response to any audio source, providing a simulated Octophonic studio experience. This paper outlines the methodology and software development process, discussing challenges and limitations of auralization with binaural impulse response. The project contributes to auralization techniques and offers a tool for musicians, engineers, and researchers interested in room acoustics and multi-channel compositions.*

1. Introduction

In the field of audio engineering and music production, it is often necessary to be present in a specific acoustic environment. However, it may not always be possible or convenient to have access to the physical space, especially in the case of a multi-channel studio with several speakers. This is where binaural methods come in, which allow us to record and experience sound as if it were being heard from human ears. Binaural impulse responses are used to emulate a space, and convolution is used to achieve a faithful reproduction of the acoustics.

The aim of this project is to discuss the development of a software, using binaural impulse responses to auralize the Octophonic studio room at Maynooth University (Figure 2). The software allows users to experience the room acoustics through headphones in a virtual environment. A specialized microphone, shaped like a human head and ear, will be used to capture the binaural impulse responses of the room. The software uses convolution to apply the captured impulse responses to any audio source, allowing users to simulate the experience of being in the Octophonic studio.

Sound localization is an important aspect of auralization, and binaural methods are particularly effective in emulating this. Interaural level differences (ILD) and interaural time differences (ITD) are key factors in sound localization, and binaural methods are capable of reproducing these effects [Begault 1994]. In this paper, we will present the methodology used to capture the binaural impulse responses of the room and the software development process. We will also discuss the importance of sound localization in auralization and the limitations and challenges associated with using binaural methods.

2. Tools and setup

2.1. Hardware and Software Tools

For this project, a variety of hardware and software tools were used. The hardware used for capturing the binaural impulse responses included the Zoom F6 recorder, and a dummy head microphone (Figure 1) which was built by Dr.Iain McCurdy, at the Music Department in Maynooth University.

The dummy head microphone was used to capture the binaural impulse response of the room. The dummy head microphone is designed to resemble the shape and size of an average human head and ears (Figure 1), allowing it to capture sound in a way that closely resembles how humans perceive sound.

In addition to the hardware tools, several software tools were used for processing and analyzing the captured impulse responses. The Reaper DAW and Logic Pro X were used for recording and editing the audio files. The Fb360 spatial audio tools and Soundfield by Rode ambisonic plugin were used for converting the ambisonic recordings to various speaker formats.

Overall, these hardware and software tools were instrumental in capturing, processing, and analyzing the ambisonic and binaural impulse responses of the Octophonic studio room at Maynooth University, and developing a software for auralization of the room acoustics.

Figure 1. Dummy-head microphone and Rode NT-SF1 microphone in the setup

2.2. The Octophonic setup

An octophonic room is a specialized listening environment that features eight speakers arranged in a circular configuration around the listener, creating an immersive audio experience (Figure 2). The listener is situated at the center of the room and can hear sounds coming from any direction, allowing for the spatial placement of audio sources [Grimshaw and Garner 2014]. This type of setup is often used in music production and sound design to create a more immersive and realistic sound field for the listener.

Figure 2. The octophonic studio in the music department at Maynooth University

3. HRIR

Head-related impulse response (HRIR), are typically represented as a collection of impulse responses measured at various positions around the listener's head [Møller 1992]. Each impulse response describes the sound's transformation from the source to the listener's left and right ears, capturing information about the time delays, spectral coloration, and spatial cues introduced by the head and ear anatomy [Møller 1992] [Blauert 1997]. By convolving an audio signal with HRIRs, it is possible to simulate how sound would be perceived by a listener in a specific location relative to their head.. [Zhang and Liu 2019]

HRIRs have found applications in a wide range of fields, including gaming, virtual reality, audio engineering, and hearing aid technology. They are essential for creating a convincing sense of auditory presence and accurately reproducing spatial sound cues, allowing listeners to perceive sound sources as coming from different directions and distances through headphones [Wightman and Kistler 1989].

4. Convolution and Deconvolution

4.1. Convolution

Convolution is a mathematical operation commonly used in digital signal processing to modify and analyze signals. It is a method of combining two functions, typically a signal and a filter, to produce a new function that describes how the input signal changes in response to the filter. The convolution operation is defined mathematically as follows:

$$f * g(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{1}$$

where f and g are two functions being convolved, and $*$ denotes the convolution operator [Lazzarini 2021]. In the context of audio processing, convolution can be used to apply a filter to a sound signal, such as a reverb effect, by convolving the signal with an impulse response of the desired filter.

4.2. Deconvolution

Deconvolution, on the other hand, is the inverse operation of convolution. It is used to extract the original input signal from a convolved signal, by dividing the convolved signal by the filter's impulse response. In audio processing, deconvolution can be used for tasks such as removing reverb from a recorded sound, or separating the effects of multiple filters from a single signal. However, it is important to note that deconvolution can be a challenging task, as it can amplify noise and other artifacts present in the original signal. Therefore, proper techniques and methods, such as regularization and truncation, should be used to minimize the impact of these artifacts. The use of these operations in binaural impulse response processing allows for the creation of a virtual acoustic space that can be experienced through headphones. [Smith 2011] [Oppenheim and Schafer 2010]

5. Sound Localization

Sound localization is the process of identifying the location of a sound source in space. Humans and other animals use different cues to determine the direction of a sound source, including inter-aural time differences, inter-aural level differences, spectral cues, and head-related transfer functions.

Inter-aural time differences (ITDs) are differences in the time of arrival of a sound wave at each ear (Figure 3) These differences are most pronounced for low-frequency sounds and can be used to determine the direction of a sound source in the horizontal plane. The brain processes these differences and uses them to calculate the azimuth angle of the sound source relative to the listener. [Bregman 1994] [Middlebrooks and Green 1991]

Inter-aural level differences (ILDs) are differences in the sound level at each ear. These differences are most pronounced for high-frequency sounds and can also be used to determine the direction of a sound source in the horizontal plane. The brain processes these differences and uses them to calculate the azimuth angle of the sound source relative to the listener. [Bregman 1994] [Middlebrooks and Green 1991]

Spectral cues are differences in the spectral content of a sound wave at each ear. These differences can be used to determine the direction of a sound source in the vertical plane. For example, a sound source above the listener will produce a different spectral content at the ears than a sound source below the listener. [Middlebrooks and Green 1991]

Head-related transfer functions (HRTFs) are filters that describe how a sound wave is modified as it travels through the listener's head and ears. HRTFs are unique to each listener and can be used to determine the direction of a sound source in both the horizontal and vertical planes. HRTFs can also provide cues about the distance and size of a sound source. [Blauert 1997]

6. Csound and Cabbage

Csound is a powerful and versatile software for sound synthesis and processing, originally developed by Barry Vercoe in the mid-1980s at the Massachusetts Institute of Technology. It uses a textual language for defining sound structures and processing algorithms, which allows for a high degree of flexibility and control over the resulting sound output. Csound provides a wide range of built-in opcodes, which are the fundamental building blocks

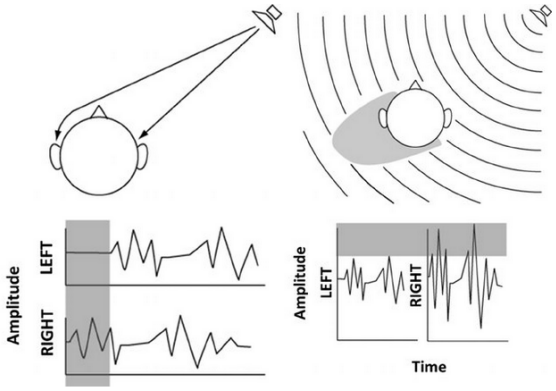


Figure 3. Sound Localization [Sun et al. 2015]

for generating and manipulating sound. These opcodes can be used to create complex synthesis techniques, including additive, subtractive, FM, granular, and physical modeling synthesis, as well as a variety of effects such as delay, reverb, distortion, and filtering. [Lazzarini et al. 2016]

Cabbage is an open-source graphical user interface (GUI) front-end for Csound, developed by Rory Walsh. Cabbage provides a modern and user-friendly interface for designing and controlling Csound instruments and effects. The interface includes various widgets and controls, such as sliders, buttons, and menus, which can be used to adjust and modulate the parameters of the Csound opcodes. Cabbage also includes a number of additional opcodes and utilities, which are not available in the standard Csound distribution, such as a sampler and a granular synthesizer. [Walsh 2015]

One of the key features of Cabbage is its ability to export Csound instruments as standalone applications or plugins for use in other software. This makes it possible to create custom instruments and effects for a wide range of applications, including music production, live performance, and interactive installations. Cabbage also supports integration with other software tools, such as Max/MSP, SuperCollider, and Pure Data, allowing for complex and flexible hybrid systems.

Csound and Cabbage were used to develop the software for this project as they offer a powerful and flexible platform for processing, with a wide range of capabilities and applications. This project exclusively uses the pconvolve opcode from csound for the convolution process.

7. Implementation

In order to auralize the room acoustics for headphones, it is essential to take into account the perception of sound by humans, especially while emulating a multi-channel sound source. To achieve this, a specialized microphone is used to capture the impulse response. The dummy head microphone, modeled in the shape of a human head with two microphones placed at each ear, replicates how a human perceives sound. However, it is impossible to create one model for all humans as each person has a different head size, shape, and size of ear, and torso. The dummy head microphone has an average-sized head, with the average size of human head circumference being 56 cm, 55 cm for females, and 57 cm for males. The average distance between the left ear and the right ear is 18 cm.

To capture a Binaural Impulse Response, a dummy head microphone is placed at the listener's position or sweet-spot. A test tone is used to measure the impulse response of the room, as it produces the maximum possible frequency for human ears over a period of time. A sine sweep of 20Hz to 20kHz is played over a time period of 7 seconds at around -12 dBFS from Room EQ Wizard (REW), a room acoustic measuring software, and recorded using Zoom F6, a 6-channel field recorder. The recordings were made in 48KHz sample rate to capture the whole frequency range of human hearing (according to the Nyquist condition). A 32-bit float bit depth was used to record, in order to avoid any unwanted distortions or clippings while recording and also for flexibility in the post-production process.

The process is repeated, and sine sweeps are played in each of the individual speakers, in this case, 8 discrete speakers and one sub-woofer for low-frequency elements. These audio files contain all binaural and spectral cues that result from the interaction of the human head with the impending sound of the source. These are referred to as head-related impulse responses (HRIRs) in the time domain or head-related transfer functions (HRTF) in the frequency domain. A head-related impulse response (HRIR) measures the magnitude (amplitude) and phase (time delay) distortions caused by different parts, like the head, size and shape of the torso, pinnae, and ear canal when sound arrives from a particular direction.

The recorded audio is then imported into Reaper, a digital audio workstation (DAW), for processing. Any DAW can be used for editing purposes. The recorded HRIRs are edited, and using a convolution plugin, ReaVerb, the distinct audio files are convolved and deconvolved (Figure 4). These HRIRs can be used to filter audio signals, resulting in a binaural signal that is equivalent to the audio signal being received from the direction from which the HRIR was recorded. This way, the room is emulated in headphones as close as how it would sound in the actual room.

This method of capturing and emulating room acoustics for headphones is extensively used in various fields like virtual reality, gaming, and music production [Chung and Kim 2019]. It can help create an immersive experience for the listener, giving the impression of being in a real room, with all its acoustical properties. It has been used in the development of many commercially successful products like Dolby Atmos, Auro 3D, and DTS:X.

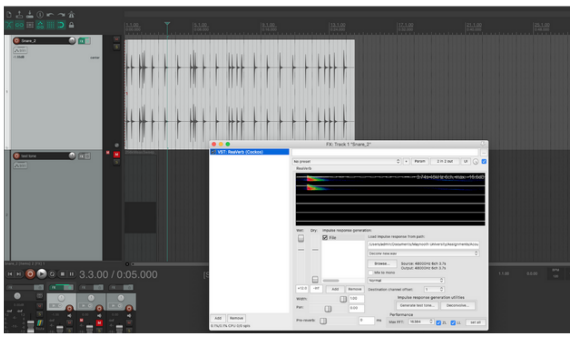


Figure 4. Screenshot of ReaVerb plugin inside Reaper DAW

8. Software System

A software application has been developed to emulate the octophonic room at Maynooth University's music department using binaural impulse response and convolution. This implementation is based on Csound and Cabbage, chosen for their benefits in providing a powerful and flexible audio programming environment.

The software application provides a user-friendly interface where the user can load a mono, stereo, or an 8-channel input audio file. The audio is then convolved using the recorded binaural impulse responses of the octophonic room, resulting in binaural audio output. The output can be listened to through headphones for an immersive audio experience.

The software application features 9 knobs, each of which controls the audio levels of a speaker in the octophonic room (Figure 5). These knobs allow the user to customize the audio output to their preference, creating a personalized audio experience. Additionally, a master fader is provided to control the overall audio level.

A meter is also available within the software application, allowing the user to visualize the audio signal. This visualization can be helpful in identifying any peaks or dips in the audio signal, aiding in audio adjustment.

Furthermore, the software application provides a "listen to source" button. This feature allows the user to listen to the original audio file before convolution, enabling a comparison between the source audio and the convolved audio.

In summary, the software application offers a powerful and flexible solution for emulating the acoustics of the octophonic room in Maynooth University's music department. It provides a user-friendly interface, customizable audio control, and helpful visualization features, making it an excellent tool for audio professionals and enthusiasts. This can be accessed from `https://github.com/joshuadanielmc/Virtual-Octophonic-Room`

Figure 5. Screenshot of the developed software using Csound and Cabbage

9. Listening Experiments

To evaluate the accuracy of the emulated octophonic room using our technique, we conducted a listening experiment with a group of 7 participants. The group consisted of 3 musicians, 3 non-musicians, and 1 participant with hearing impairment. In the experiment, a snare sound convolved with the Head-Related Impulse Responses (HRIRs) was

played back 10 times from different speakers among the 8 speakers emulated in the octophonic room. The participants were then asked to localize the sound using headphones.

The results of the experiment showed that the participants were able to accurately localize the sound in the emulated octophonic room. The localization was consistent across all the participants, including the participant with hearing impairment. This indicates that the emulated room was able to reproduce the spatial characteristics of the original octophonic room with a high degree of accuracy.

The listening experiment was conducted in accordance with the ethical guidelines for research involving human subjects. All participants gave their informed consent to participate in the experiment. The study was approved by the Institutional Review Board of the University.

The results of the experiment demonstrate the effectiveness of our technique in emulating real-world acoustic spaces with a high degree of accuracy. This has practical applications in a range of fields, including music production, virtual reality, and architectural acoustics.

Table 1. Listening Experiment Results

Test Subjects	No. of Correct answers
Musician A	7
Musician B	7
Musician C	8
Non-Musician A	8
Non-Musician B	7
Non-Musician C	6
Hearing Impaired Subject	6

10. Practical Applications

This system has several practical applications and commercial values that can be implemented in real-life scenarios. One such use is in virtual reality applications. The system the system could be extended to be used for musical applications to emulate different studio spaces worldwide. For instance, if a user wants to listen to their mixes in a specific studio space and speakers, they can capture the head-related impulse response of that studio and play back their audio through this system. Similarly, this system can be used to test mixes in concert spaces or performance venues before the actual performance. A mix that has been played in one's studio might differ when being played back in a performance venue. By capturing the impulse response of the venue or concert hall and sending it to the performers or composers, they can have an idea of how their music will sound at the venue and modify their piece accordingly.

Furthermore, there is a significant scope in meditation purposes. Using this system, one can emulate different spaces and use them for meditation purposes. Since impulse responses are captured for the auralization process, the data can be used to analyze the room's acoustics and address any acoustical problems. The same data can also be used to modify the room's acoustical properties using physical elements like absorbers, diffusers, or digitally.

In terms of commercial value, this system can be used by audio engineers, music producers, and composers to check their mixes in different acoustic spaces. This can save time and resources by eliminating the need to physically visit different studios and venues. Additionally, this system can be used in the education and research fields to teach acoustics and sound engineering. This system's ability to auralize different acoustic spaces can help students understand the effects of different room acoustics on sound.

11. Existing systems

The developed system offers a distinct approach in the digital audio processing field, setting it apart from exisiting systems like Slate Digital's Slate VSX [Digital 2023]. Unlike Slate VSX, the developed system is open-source, adaptable to various headphones, more budget-friendly, and supports multi-channel audio. It can also be used as standalone software, addressing several limitations of Slate VSX, making it an appealing choice for a wider range of users in the audio processing domain.

Although Slate VSX has some limitations, it has proven to be a popular system for emulating different studio spaces and headphone listening experiences. Our system builds upon the limitations of Slate VSX by offering a more cost-effective solution that is not tied to specific hardware and supports multi-channel audio.

12. Limitations

In order to provide a comprehensive understanding of the system, it is essential to discuss the limitations that our system faces. The first limitation is related to the physical differences of human beings. Each person has a unique shape of the ear, different head size, and different torso, which makes it challenging to develop a system that will auralize sound accurately for everyone. Thus, the system's accuracy is limited by the variation in the listeners' physical characteristics.

Another limitation is that binaural audio can only be played back using headphones. Although this allows for an immersive audio experience, it also means that the system is limited to headphone playback, which may not be ideal in some situations.

Finally, the listener's position and head movement are in a fixed position while listening to the binaural audio, which limits the experience that we get in a physical space. In contrast, in a physical space, we have the freedom to move our head and change our position, which can affect the sound's perceived direction and spatial characteristics. Therefore, our system's limitations should be taken into account when considering its potential applications.

13. Further Research

The system we have developed has a lot of potential for further research and development. Some of the future scopes for this project include upgrading the system with more inputs and multi-channel output, allowing for a wider range of applications beyond just binaural. Additionally, implementing a calibration process in the chain could make it possible to use any headphones with the system, rather than being limited to specific hardware. The method for headphone compensation is presented for Neumann KU 100 is already introduced in [Bernschütz 2013], and the same method can be applied in this system.

Another area for further exploration is creating both standalone and plugin versions of the software, which would allow users to use the system without needing to open the cabbage application. Finally, implementation of ambisonics and other spatial audio formats could further improve the system and make it even more versatile for different applications.

Another future work is also to explore low-order IIR models for the BIR dataset, which may be a viable and computationally lightweight alternative.

These areas represent exciting possibilities for the future of this technology, and we look forward to seeing how it evolves in the years to come.

14. Acknowledgement

I would like to express my sincere gratitude to Dr. Iain McCurdy, who supervised the acoustic and psychoacoustic module of our course and provided invaluable guidance and support throughout the project. His expertise knowledge in the field of acoustics and psychoacoustics, and Csound has been instrumental in the success of our project.

I would also like to thank Prof. Victor Lazzarini for his guidance and teaching during the course. His knowledge and experience in the field of digital signal processing, and software programming have been invaluable in shaping this project.

References

Begault, D. R. (1994). Binaural technology: Getting the sound where it counts. *Audio*, 78(8):42–47.

Bernschütz, B. (2013). A spherical far field hrir/hrtf compilation of the neumann ku 100. In *Proceedings of the 40th Italian (AIA) annual conference on acoustics and the 39th German annual conference on acoustics (DAGA) conference on acoustics*, volume 29. German Acoustical Society (DEGA) Berlin.

Blauert, J. (1997). *Spatial hearing: The psychophysics of human sound localization*. MIT press.

Bregman, A. S. (1994). *Auditory scene analysis: The perceptual organization of sound*. MIT Press.

Chung, E. and Kim, Y. (2019). A study on binaural impulse response analysis and its application for virtual reality audio. *Applied Sciences*, 9(11):2321.

Digital, S. (2023). Slate vsx. `https://www.slatedigital.com/slate-vsx-virtual-mix-room/`. Accessed: 2023-04-06.

Grimshaw, M. and Garner, T., editors (2014). *The Oxford Handbook of Sound Studies*. Oxford University Press.

Lazzarini, V. (2021). *Spectral Music Design: A Computational Approach*. Oxford University Press.

Lazzarini, V., Yi, S., Heintz, J., Brandtsegg, Ø., McCurdy, I., et al. (2016). *Csound: A sound and music computing system*. Springer.

Middlebrooks, J. C. and Green, D. M. (1991). Sound localization by human listeners. *Annual Review of Psychology*, 42(1):135–159.

Møller, H. (1992). Fundamentals of binaural technology. *Applied acoustics*, 36(3-4):171–218.

Oppenheim, A. V. and Schafer, R. W. (2010). *Discrete-time signal processing*. Pearson, 3rd edition.

Smith, J. O. (2011). *Spectral Audio Signal Processing*. W3K Publishing.

Sun, L., Zhong, X., and Yost, W. (2015). Dynamic binaural sound source localization with interaural time difference cues: Artificial listeners. *The Journal of the Acoustical Society of America*, 137(4):2226–2226.

Walsh, R. (2015). Cabbage: A desktop environment for csound. In *Proceedings of the 12th Sound and Music Computing Conference (SMC 2015)*, pages 307–313. Maynooth University.

Wightman, F. L. and Kistler, D. J. (1989). Headphone simulation of free-field listening. i: Stimulus synthesis. *The Journal of the Acoustical Society of America*, 85(2):858–867.

Zhang, Y. and Liu, Y. (2019). Binaural impulse response (bir). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27:1367–1379.

Creative Possibilities and Customizability of Live Performance Systems with Open Source Programming Platforms.

Aman Jagwani

Department of Music, Maynooth University

Abstract. *A vast array of open source programming platforms exists in the area of audio and music programming that can enable musicians, composers and programmers to create custom devices that can be used in live performance situations. This process can extend the creative involvement of the artist into the manifestation of the performance system itself, enabling better customization of the final artistic results towards their creative vision. This paper will present discussions, examples and ideas for devices that can be created with some of these platforms across both desktop and embedded environments and how these devices can be used to add a greater degree of creative expression during live performance. The paper will conclude with an overview of an on-going project to develop a similar, accessible platform for FPGAs.*

1. Introduction

With the advent of modern control systems such as advanced grid controllers like the Ableton Push 2 [Ableton 2015], responsive and articulate gestural controllers like the MiMU gloves [MIMU 2019] and Midi polyphonic expression [Association 2020] based controllers like the Roli seaboard [Roli 2022], the integration of electronics in live performance has been widely adopted to great effect. Across different musical idioms such as computer music, electro-acoustic music, jazz, fusion and commercial music, the integration of modern control systems has added the possibility of timbral, gestural and sonic exploration to live performance with the use of electronics. However, the fact that most commercially available control systems are generalized and provide a limited degree of customizability, a certain amount of adaptation and possibly compromise may be required from the artist in order to effectively use these systems. Furthermore since these systems are limited to just controlling devices such as synthesizers and effects processors, their utility is limited in being an interface to other tools.

Open source programming platforms help mitigate these problems. These platforms allow users to design their own devices and control systems, enabling them to tailor their entire performance environment to their musical and sonic needs and limitations. In a way, these platforms extend the creative or compositional process one step back into the instrument design process. If artists are able to express their creative voice into the creation of their tools, they may be able to manifest their artistic vision to a greater extent. Furthermore, the fact that these platforms can be used to customize both hardware and software aspects of performance systems means that artists are able to create the most effective system in terms of interaction as well as musical and sonic output.

This paper will discuss some of these platforms for both desktop and embedded environments. It will also present some examples of devices and systems created with

each of these platforms that highlight their effectiveness in live performance settings. Additionally future development possibilities, particularly with respect to the use of FPGAs will complement the discussion.

Sections of code will be provided to highligh key aspects of the examples below. Complete code will be available in this repository:

https://github.com/amanjagwani/Live-Performance-Devices.git

2. Desktop Platforms

There are a large variety of open source sound and music programming languages such as Csound, Supercollider and Faust to name a few. This paper will focus on Csound, particularly on the use of Csound within the environments of Cabbage and Python and how these tools can facilitate the seamless integration of custom devices in live performance systems.

2.1. Csound in Cabbage

Csound is a sound and music computing system. It is one of the most extensive tools for electro-acoustic composition and sound design [Lazzarini et al. 2016].With the advent of modern front ends and frameworks for Csound as well as the release of the Csound host API, the use of Csound has expanded into the realms of plug-in development, interactive systems and live performance tools. One such tool-set is the Cabbage front end. Developed by Rory Walsh, Cabbage is a framework for audio software development using the Csound language. It allows users to develop Csound based software across various platforms such as VST, Audio Unit, android etc [Walsh 2008].

The examples in this section will focus on VST plug-ins made with cabbage and particularly on how the features of both Cabbage and Csound can be used to enhance live performance systems. The use of this platform can lend itself well to three categories of use:

2.1.1. Performance Utility and Flow

An example of how Csound in Cabbage can be used to support the flow of performance is a musical percussive trigger plug-in. There are many commercial options for percussive trigger systems available in the market that utilize sensors like piezo discs to detect impulses and subsequently trigger drum samples. However, most of these systems can be limited in their musical involvement beyond triggering pre-defined sounds. With Csound in Cabbage, a trigger system can have a larger variety of musical responses. With this platform an example trigger system can be developed that triggers a sequencer controlling an FM synth to run for a particular amount of time. The velocity of the percussive gesture can define the amount of time the sequencer runs for as well as the FM index. With cabbage's in-built ability to add tempo synchronization functionality to any project, this plug-in can be integrated well within a Digital Audio Workstation or any other host, generating tempo-synced melodic sequences concurrently with percussive performance. This utility plug-in can enable the performer to maintain the flow of percussive performance while bridging the gap between the rhythmic, sonic and melodic aspects of a performance through musically involved responses.

2.1.2. Musical, Timbral and Sonic Exploration

Csound provides numerous opcodes or unit generators for creative real-time sound design. A particularly great set of opcodes for this purpose are the Phase Vocoder (pvs) opcodes. These opcodes enable a variety of sonic transformations in the frequency domain [Lazzarini et al. 2006]. Some of these opcodes include pvsbuffer, pvsblur, pvshift, pvsmorph, pvscale etc. The presence of such a large variety of individual spectral unit generators implies that a modular approach can be taken to implement effects such as spectral delay, blurring, morphing and scaling to create unique combinations of sound transformations [Lazzarini 2021]. Furthermore, the fact that these opcodes are optimized for fast, real-time performance means that they can be very beneficial in live performance situations, especially in combination with live gestural control.

An example of a performance system that makes use of the above features is a modular spectral effects plug-in made with Cabbage. This plug-in consists of spectral delay implemented with the pvsbuffer and pvsbufread opcodes, blurring with pvsblur and pitchshifting with pvscale. The combination of these opcodes can create an extremely large variety of sound transformations. The pvsbuffer and pvsbufread themselves can create many effects such as time scaling by adjusting the buffer read rate, glitching by adding randomness to the buffer reading as well as granular or arpeggiation effects by using short delay times with feedback. By adding spectral blurring and pitch scaling, the transformation options are increased. Additionally, periodic freezing can be implemented by sampling and holding between frozen and unfrozen read pointers of the buffer, adding a rhythmic dimension to the sound transformations. When used in live performance situations, this device can be very effective in creating dramatic or subtle sound transformations.

The code snippet below highlights the modular approach to using pvs opcodes in Csound:

```
fbuf pvsbufread kpoint - kdelay, kh
fbp pvsbandp fbuf, kmin, kmin+150, kmax, kmax-150
fscale pvscale fbp, kscale
fblur pvsblur fscale, kblurtime, 1
aresyn pvsynth fblur
```

2.1.3. A combination of Utility and Musicality

The use of this platform is also conducive to the creation of tools that perform both utility and musical functions in live performances. One example for this is a Transition-Maker Plug-in. Often in performance situations, performers need to make a sonic or musical statement to move across consecutive sections or mark milestones in a musical form. This can be done by triggering pre-recorded transition sound samples or with momentary sound transformations. The problem with the first option is that these samples can sound generic and can create a lack of musical variety or interest. The problem with the second option can be that purposeful and effective sound transformations require considerable performative involvement, possibly breaking the flow of the performer. Furthermore, these transitions require a large degree of timing accuracy and often have to be carried

out very quickly. To solve these problems, a transition-maker plug-in is a good example. This plug-in carries out granular processing, modulated filtering and cross-synthesis of live audio with filtered pink noise to create a unique time-synchronised transition every time it is triggered. All of the processing is controlled by envelope generators and random modulators over a specified number of beats in the host application's tempo, creating effective sonic output for the purpose of transitions. With the in built support for MIDI provided by Csound, the plug-in can be controlled by a simple note-on message, enabling performers to be able to quickly instantiate a unique transition during live performance without breaking their performative flow.

This code snippet shows how the cross synthesis part of the plug-in is implemented over the specified duration of beats it receives via p3:

```
knoisefilt linseg 1000, p3/2, 10000, p3/2, 12000
ains = (al+ar)*0.5
knoise linseg 0, p3/2, 1
anoise pinker
anoise = anoise*knoise
anoise moogladder anoise, knoisefilt, 0.4
kamp1 rspline 0.1, 1, 0.3, 10
kamp2 = 1-kamp1

finput pvsanal ains, 512, 128, 512, 1
fnoise pvsanal anoise, 512, 128, 512, 1
fcross pvscross finput, fnoise, kamp1, kamp2
across pvsynth fcross
kcrossamp linseg 0.2, p3, 1
across = across*0.2*kcrossamp*knoiselvl
```

2.2. Csound in Python

With the Csound API, users are able to easily incorporate Csound into their own applications. The API allows users to harness the powers of Csound within different environments such as C++, Javascript, Python etc., through a serious of different functions [Lazzarini et al. 2016]. In Python, the ctcsound module, created by Francois Pinot in 2015, based on the Python Ctypes library, provides a convenient interface for using Csound within Python [The Csound Development Team 2021]. This means that features of Python and the audio programming capabilities of Csound can be combined to create unique systems for live performance.

One feature of Python that is worth talking about in this context is how conducive it is for computer vision and machine learning applications. Keeping this and the fact that the ctcsound module enables multi-threading in mind, a hand-tracking based real-time Csound controller can be made. Using the cv2 and mediapipe modules in python, hand gestures can be tracked from real-time camera input [Google 2021]. Some examples of these are x or y position of the hand, hand movement velocity, openness of the hand, hand actions such as thumbs up etc. The values generated from these gestures can then be mapped to various musical or sonic parameters in Csound, creating an interface for hand-movement based interaction with Csound. With the Csound audio engine running on a separate thread, real-time audio accuracy is maintained even while this complex chain of interaction and analysis is carried out in the main thread.

Below is a section of code that uses hand-tracking to control Csound in real time. In this example, x position of the hand is mapped to timescaling, y position is mapped to pitch-scaling, hand velocity is mapped to granular reverb amount and hand openness is mapped to grain size.

```
#loop for hand tracking
while True:
    success, image = cap.read()
    image = cv2.flip(image, 1)
    results = process_image(image)
    draw_hand_connections(image, results)
    if results.multi_hand_landmarks:
        #get y position of hand
        hand_y = results.multi_hand_landmarks[0].landmark[0].y
        #get x position of hand
        hand_x = results.multi_hand_landmarks[0].landmark[0].x
        #get openness of hand
        hand_openness = results.multi_hand_landmarks[0]
                        .landmark[0].y -
                        results.multi_hand_landmarks[0]
                        .landmark[8].y
        #hand velocity calculation
        if prev_results:
            time_delta = time.time() - prev_time
            index_x = results.multi_hand_landmarks[0].landmark[8].x
            index_y = results.multi_hand_landmarks[0].landmark[8].y
            prev_index_pos = (prev_results.multi_hand_landmarks[0]
            .landmark[8].x,
            prev_results.multi_hand_landmarks[0].landmark[8].y)
            curr_index_pos = (results.multi_hand_landmarks[0]
            .landmark[8].x,
            results.multi_hand_landmarks[0].landmark[8].y)
            index_vel = calculate_velocity(prev_index_pos,
                                curr_index_pos, time_delta)
            #mapping hand velocity to amount of
            #granularized reverb
            verb = scale(index_vel, 0, 5, 0, 0.8)
            #setting control channel
            cs.setControlChannel('granvol', verb)
        prev_results = results   #update state
        prev_time = time.time()
        #mapping hand params to csound params and
        #setting control channels
        pitchshift = scale(hand_y, 1, 0, 1, 8)
        timescale = scale(hand_x, 0, 1, 0.05, 4)
        grainsize = scale(abs(hand_openness),
                    0, 0.6, 0.05, 0.5)
        cs.setControlChannel('pitchshift', pitchshift)
        cs.setControlChannel('timescale', timescale)
        cs.setControlChannel('gransize', grainsize)
        cv2.imshow("Hand␣Tracker", image)
        #stop videocapture and close window if the 'q'
        #key is pressed
        if cv2.waitKey(1) == ord('q'):
        break
```

```
cap.release()
cv2.destroyAllWindows()
```

3. Embedded Audio Platforms

Open source embedded platforms for audio and music programming enable makers and artists to create custom devices that can support their musical goals. Embedded platforms add various advantages such as portability and interaction design with the use of sensors. Moreover, once again they extend the creative process into the manifestation of the performance tool itself, allowing the artist to completely streamline their performance system to their vision. After the rise of the Arduino, a variety of open source embedded audio programming platforms have been emerging. Similar to the Arduino, many of these platforms have enabled the accessibility of programming hardware devices such as micro controllers by providing high level interfaces to complex hardware programming. Some examples of these are the Daisy platform that allows the programming of an STM32 based microcontroller board, the Bela board which provides similar accessibility to the BeagleBone Black board and the faust2teensy command line application for programming the Teensy Board with Faust [Michon et al. 2019]. This section will focus on the use of the Daisy and Bela platforms and present examples of how they can play specific, customized roles in live performance situations.

3.1. The Daisy Platform

This is an open source embedded audio programming platform developed by Electrosmith. It is based on the STM32 microcontroller and provides all peripherals required for audio applications such as a 24 bit, 96khz audio codec, digital and analog IO, multiple memory locations, power and support for various communication protocols (UART, SPI, I2C etc) on a single compact board [Electro-Smith 4 24]. The daisy provides a higher level abstraction of the STM32 HAL through libDaisy, a hardware support library that implements communication with the Audio Codec and takes care of the hardware peripherals, enabling the user to focus on the creative and musical aspects of coding and design with this platform [Electro-Smith 4 24].

3.1.1. Daisy Example: Flanger, Pitchshifter Effect

The Daisy platform provides support for programming with Pure Data, Max Msp's gen , Arduino and C++. This example was created with C++ on specifically the Daisy Seed which is a system on module(SOM) for embedded audio applications. It is extremely compact, without any connectors on board and can fit into various embedded applications such as synth modules, guitar pedals or breadboards. Electrosmith provides various breakout boards centered around the Daisy seed for various platforms such as Eurorack and guitar pedals. However, anything created with the seed is transferable to any of these since the differences are merely the interface to the seed. This example implements a real time Flanger and Pitchshifter effect with the use of variable delay lines. The fact that a change in delay time translates to changes in pitch due to the difference in the delay line read and write rates is at the center of both these effects and can be viewed in the code snippet below. The flanger effect is implemented first by using a low frequency oscillator

to modulate a variable delay line. Then the output from this effect is fed into the pitch shifting algorithm that works by creating a continuous rate of change of the delay time to cause a static change in pitch of the sound.

3.1.2. The Aurora DSP Library

Electrosmith provides DaisySP, a C++ library of DSP objects ported from other open source platforms However, for this example, the Aurora library by Victor Lazzarini was used. Aurora is an open source, minimal, lightweight, header-only C++ audio synthesis and processing toolkit. This library has no external dependencies and no linking is required to use it since solely the headers of DSP objects that are used in a program are required to be included. Furthermore, the library covers most commonly used synthesis and processing techniques, accessible very seamlessly, in a generic manner through templating [Lazzarini and Walsh 2023]. These factors make it ideal for use with embedded systems.

Here is a section of code entailing the processing with Aurora objects in the Audio Callback Function:

```
void AudioCallback(AudioHandle::InputBuffer in,
    AudioHandle::OutputBuffer out, size_t size)
{
  // get control values from analog ins
  fr = (hw.adc.GetFloat(0)) * 10;
  fb = hw.adc.GetFloat(1);
  kp = hw.adc.GetFloat(2)*8;
  mixlvl = hw.adc.GetFloat(3);
  kf = -(kp-1)/mxdel;
  bufferL.resize(size);
  bufferR.resize(size);
  lfoL.vsize(size);
  lfoR.vsize(size);

  // Copy input to processing buffer
  std::copy(in[0], in[0] + size, bufferL.begin());
  std::copy(in[1], in[1] + size, bufferR.begin());

  //Process audio
  const std::vector<float>& flangeL = delayL(bufferL,
     admxL(gainL(lfoL(0.005, fr), flangain), deltL), fb);
  const std::vector<float>& flangeR = delayR(bufferR,
     admxR(gainR(lfoR(0.005, fr+1), flangain), deltR), fb);

  const std::vector<float>& wetL = adpsL(gwL1(pdelayL1(flangeL,
     lphs1(mxdel, kf, 0), fb), lw1(1, kf, 0)), gwL2(pdelayL2(flangeL,
     lphs2(mxdel, kf, 0.5), fb), lw2(1, kf, 0.5)));
  const std::vector<float>& wetR = adpsR(gwR1(pdelayR1(flangeR,
     rphs1(mxdel, kf, 0), fb), rw1(1, kf, 0)), gwR2(pdelayR2(flangeR,
     rphs2(mxdel, kf, 0.5), fb), rw2(1, kf, 0.5)));

  const std::vector<float>& outL = addrywetL(drylvlL(bufferL, mixlvl),
     wetlvlL(wetL, 1-mixlvl));
```

```
  const std::vector<float>& outR = addrywetR(drylvlR(bufferR, mixlvl),
      wetlvlR(wetR, 1-mixlvl));

  // Copy output from processing to output buffer
  std::copy(outL.begin() , outL.end(),  out[0]);
  std::copy(outR.begin() , outR.end(),  out[1]);
}
```

The aspect of this example that highlights the benefits of open source embedded audio programming platforms is that despite being readily available, common effects, this specific combination of these processing techniques may be unique. Similarly, users can create their own specific combination of processing or synthesis techniques that works best for their musical goals. Furthermore with the use of different options for analog control, the live interaction interface for the effects can also be customized to fit the users' performance requirements.

3.2. The Bela Platform

The Bela is an open-source embedded platform for audio programming. The Bela is based on a Beagle Bone Black single board computer. Therefore, in contrast to the bare metal Daisy platform, the Bela is an example of an embedded linux platform. However, the Bela consists of a custom hard real-time audio environment based on Xenomai Linux that provides extremely low audio latencies. Similar to the Daisy, the Bela provides stereo audio input and output (with the option of a multi-channel expander cape), 8 channels of analog inputs and output and 16 digital I/Os [McPherson 2017].

The Bela provides support for programming with many different languages including C++, Pure Data, SuperCollider and Csound. To contrast the example with the Daisy, the example below will use Csound to program the Bela, highlighting the possibility of even higher level programming for embedded applications.

3.2.1. Bela Example: Moog DFAM-based Drum Machine

The Moog DFAM is a desktop creative drum pattern generator and synthesizer. It has a large number of useful sound design features such as pitch, filter and amplitude envelope generators, frequency modulation, waveform selection, noise generators etc. Additionally it has two 8 step sequencers to control pitch and velocity (this sequence also affects the amount of each of the envelopes applied to the signal). The presence of these features along with the compact form factor and intuitive design makes it possible to achieve very powerful sonic results relatively easily with this instrument.

The Bela version of this instrument models all of the sound design features of the original but also harnesses the powers of Csound and Bela. The original instrument consists of over 30 potentiometers to control the parameters. This amount of controls is too much for the standard Bela with the limited number of analog inputs available. However, this problem presents the opportunity to replace a lot of these controls with Csound's comprehensive collection of modulation sources and random number generators. Parameters such as envelope amounts, noise amounts, frequency modulation, cutoff frequency etc can be modulated by opcodes such as rspline and jspline in Csound, adding a semi-generative quality to the instrument. Furthermore, the original instrument has a patch bay

to add additional modulations and CV routings to the instrument. Since this is difficult to implement again with the limited analog inputs and outputs of the Bela, an alternative sonic extension to the instrument can be Csound's signal processing capabilities. Some signal processing methods used in the Bela implementation are:

Adaptive Frequency Modulation - As described in this paper by Victor Lazzarini, Joseph Timoney and Thomas Lysaght [Lazzarini et al. 2007], this technique is based on frequency or phase modulation of arbitrary input signals. It is implemented on the basis of frequency modulation effects caused by periodically varying the delay time of a delay line at audio rates while passing the input signal through it. This effect is useful in adding interesting high frequency excitation to the synthesized drum sounds.

Sample rate folding - This is a commonly used effect to introduce low fidelity artifacts to input signals. It is particularly effective with drum signals especially when the amount of folding is modulated over time.

Tempo Synced Echo - This effect can add some metric variations to the drum pattern by modulating the delay time in sync with the tempo of the pattern. The use of a resonant filter within the delay line also adds some timbral variation this.

Automating parameter modulation with rspline opcode:

```
kfm rspline 0, 1, 0.1, 4
kcf rspline 200, 12000, 0.1, 4
knoisemod rspline 0, 1, 0.1, 4
kpchdecay rspline 0.1, 4, 0.1, 4
kampdecay rspline 1, 4, 0.1, 4
```

Receiving control values through analog ins and some of the effects processing:

```
ain chnget "fxSend"
amod chnget "analogIn2"
kmod = k(amod)
kmod scale kmod, 0, 1, 0, 0.3
//Additive FM Processing via an AdFm UDO
kindex = kmod*15
afm AdFM ain, 4, kindex
afm butterlp afm, 18000
afm ntrpol ain, afm, 0.5
// sample rate folding
afold fold ain, 1+kmod*100
```

The combination of the above three effects creates unique and interesting sound transformation possibilities, especially with gestural control and modulation. The possibility of using sensors with the Bela platform is very beneficial for this purpose. In this implementation, a Force Sensitive Resistor can be used to control the amount of effects being applied to the drum synth and a light dependent resistor can be used to control a modulation parameter which is mapped to the index of adaptive FM, the amount of sample rate folding, the tempo synced delay time as well as the cutoff frequency of the delayed signal. The two sensors allow the performer to interact with the timbre of the drum pattern with intuitive gestures and create interesting real time sound transformations during live performances. This combination of synthesis, processing and interaction enables unique musical performance possibilities with an instrument that is tailored to a

specific artistic vision, highlighting once again, the benefits of customizability that open source embedded audio programming platforms provide.

4. Future Work: An Open Source Programming Library for FPGAs

FPGAs or field programmable gate arrays are integrated circuits consisting of a large array of configurable logic cells [Kastner et al. 2018]. This definition implies that they can be configured to create a variety of digital circuits and thus can achieve almost any computational task. With the strong possibilities of parallelization, high throughput, extremely low latency and high sample rates, these chips are highly conducive to audio processing applications [Risset et al. 2020]. However due to the high complexity and involvement in low level hardware design required for programming FPGAs they are not very accessible to the general creative audio and music community. A current work-in-progress is the creation of an audio DSP programming library of hardware intellectual property(IP) blocks or modules to streamline the workflow of audio programming for an FPGA with the potential to make this programming environment more accessible.

Traditionally, FPGAs are programmed using Hardware Description Languages such as VHDL and Verilog. These languages require extremely specialized hardware design knowledge. To provide a wider range of users with the ability to program FPGAs, vendors such as Xilinx and Intel have provided High Level Synthesis (HLS) tools. These tools allow users to create a functional specification for a hardware design module in a (comparatively) higher level language such as C or C++ that will be converted into an RTL hardware design by the tool which in turn can be synthesized into a bitstream that can be flashed onto an FPGA [Kastner et al. 2018] .

Although HLS tools provide higher level access to FPGA programming, they still require a high degree of hardware design knowledge as well as knowledge of the various vendor specific tools that are required to be used to program FPGAs. This still results in these environments remaining largely inaccessible to the general creative audio and music programming community. The goal of this project is to bridge this gap by providing a platform that users with audio programming knowledge in C or C++ can use to create their own FPGA based audio and music devices. This project will also involve the creation of custom synthesis device examples with the platform that showcase the powers of FPGAs and the benefits of the platform in creating customized devices for live performance.

Within the scope of this project, a modular sound synthesis platform has been made. The HLS programming flow generally entails the design of a top level function that gets translated into a standalone hardware IP (intellectual property) module with its arguments defining inputs and outputs. Within the function, along with processing algorithms, optimization directives or pragmas have to be used for behavior, communication and interface definition. The generated IP module has to then be integrated and deployed in a complete hardware design using a set of downstream FPGA design tools such as Vivado and Vitis in the case of the AMD/Xilinx FPGA environment [Xilinx 2023]. Within this context, the modular platform consists of a set of inter-connectable sound synthesis HLS IP cores or modules, accompanied by a base audio system to exemplify the feasibility and design considerations of modular synthesis on FPGAs. The modules include different types of oscillators, filters, envelope generators and other components of modular synths. These can be combined to perform various different types of sound synthesis

such as subtractive, additive and FM synthesis.

A similar project that provides even higher level access to FPGA programming is the Syfala project by the Emeraude team with researchers from INRIA, GRAME and INSA, in Lyon, France. This is a tool chain that allows users to program DSP algorithms using the Faust programming language [Popoff et al. 2022].

5. Conclusion

This paper discussed the use of several open source audio and music programming platforms to enhance and customize live performance systems. Platforms across both embedded and desktop environments were covered and the examples presented for each of the platforms highlighted how live performance systems can be enhanced with the creation of an artist's own performance tools and also how the creative vision of the artist can be manifested very powerfully with the help of these tools. Lastly, an on going project for developing a similar platform for FPGAs was presented.

References

Ableton (2015). Ableton push. `https://www.ableton.com/en/push/`.

Association, T. M. (2020). Midi polyphonic expression (mpe). `https://www.midi.org/midi-articles/midi-polyphonic-expression-mpe`.

Electro-Smith (Accessed: 2023-04-24). Daisy: An Embedded Platform for Music. `https://www.electro-smith.com/daisy/daisy`.

Google (2021). Mediapipe: Cross-platform framework for building multi-modal machine learning applications. `https://developers.google.com/mediapipe`. [Online; accessed 2-April-2023].

Kastner, R., Matai, J., and Neuendorffer, S. (2018). Parallel programming for fpgas. *arXiv preprint arXiv:1805.03648*.

Lazzarini, V. (2021). *Spectral music design : a computational approach*. Oxford University Press, New York, NY.

Lazzarini, V., Lysaght, T., and Timoney, J. (2006). Spectral signal processing in csound 5. *International Computer Music Association Proceedings*, 2006:392–395.

Lazzarini, V., Timoney, J., and Lysaght, T. (2007). Adaptive fm synthesis. In *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07)*, volume 1, pages 305–312.

Lazzarini, V. and Walsh, R. (2023). Aurora-lattice: Rapid prototyping and development of music processing applications. Unpublished manuscript.

Lazzarini, V., Yi, S., ffitch, J., Heintz, J., Brandtsegg, Ø., and McCurdy, I. (2016). *Csound: A Sound and Music Computing System*. Springer.

McPherson, A. (2017). Bela: An embedded platform for low-latency feedback control of sound. *The Journal of the Acoustical Society of America*, 141($5_{S}upplement$) : $3618 - -3618$.

Michon, R., Orlarey, Y., Letz, S., and Fober, D. (2019). Real time audio digital signal processing with faust and the teensy. In *Proceedings of the Sound and Music Computing Conference*, pages 325–332, Malta.

MIMU (2019). Mimu gloves. https://mimugloves.com/.

Popoff, M., Michon, R., Risset, T., Orlarey, Y., and Letz, S. (2022). Towards an fpga-based compilation flow for ultra-low latency audio signal processing. In *SMC-22 - Sound and Music Computing*, Saint-Étienne, France.

Risset, T., Michon, R., Orlarey, Y., Letz, S., Muller, G., È., and Gbadamosi, A. (2020). afaust2fpga for ultra-low audio latency: Preliminary work in the syfala project. In *Proceedings of the 2nd International Faust Conference*, page 10.

Roli (2022). Roli seaboard rise. https://roli.com/products/seaboard/rise2.

The Csound Development Team (2021). CTCSound API Documentation. https://csound.com/docs/ctcsound/ctcsound-API.html. [Online; accessed 2-April-2023].

Walsh, R. (2008). Cabbage, a new gui framework for csound.

Xilinx (2023). Vitis hls user guide. Accessed: 2023-01-18.

Paper-Arts Practice

Live sampling and improvisation in iubar project's participatory music performance

Martin K. Koszolko
The University of Newcastle, Australia
Martin.Koszolko@newcastle.edu.au

Abstract. *This document presents an artistic proposal for a participatory live performance aiming to explore improvisation and comprovisation techniques while investigating audience agency in electronic performance settings. The performance is featured online at UbiMus23 and includes edited footage and audio recordings from the iubar project's participatory live music performance at Hashtag Lab in Warsaw, Poland, held on July 4th, 2023.*

1. Background

Iubar project is Martin Koszolko's solo act focusing on live sampling, story-driven ambient, electroacoustic music and incorporating field recordings. The performance at Hashtag Lab (2023) aligns with the UbiMus23 theme of improvisation and comprovisation in ubiquitous music. It is an outcome-focused performance that follows Breel's performative approach, wherein "participants respond with individual behaviour whilst the performance as a whole remains within the original parameters established by the artist" (2015).

This performance aims to raise questions about the feasibility of applied improvisation and comprovisation techniques, exploring their impact on audience engagement and agency. This exploration follows a 3-stage process, building upon prior research by the author in the realm of live sampling and improvisation utilising mobile music technologies (Koszolko 2021, 2022).

Additionally, the performance integrates selected comprovisational techniques (Borgo 2022) to address and challenge predetermined power relations between musicians and the audience. By employing improvisation based on unique audience samples obtained on location, it seeks to overcome the limitations often present in Western art music "written by a composer and performed from a score, which normally leaves no room for participatory creativity by audience members" (Toelle & Sloboda 2021).

2. Performance Description

For UbiMus23, the original 1-hour long performance has been condensed into a 15-minute video excerpt with subtitles and additional captions delineating various event stages. The edited video includes music fragments and audience interactions.

The performance process is organised around 3 key stages:

1. Live sampling: The event begins with posing a question as the initial impulse and recording audience responses.
2. Improvised Electronic Live Performance: The recorded audience samples are the foundation for this stage, divided into two distinct stylistic sections (A and

B). The first section delves into ambient music, while the second explores a more rhythmic electronica style.

3. Voting: The audience actively participates by voting on their preference between section A and B.

In stage 1, the performer introduces the general approach and concept of live sampling, followed by an invitation for audience participation. The audience is prompted to respond to a series of questions, which at the Hashtag Lab event revolved around the theme of 'collectively or individually'. Live sampling of audience voices responding to these posed questions forms a crucial component.

After a brief period of editing the recorded material, the performance unfolds over approximately 50 minutes. Collected audio samples are synthesized, sequenced, and processed live by the performer using mobile devices, specifically 2 iPads and 1 iPhone, employing a range of iOS apps. These apps include AUM – Audio Mixer, Koala Sampler, DrumComputer, Shoom Synthesiser, ThumbJam, DrumJam, Mononoke, Fluss – Granular Playground, GlitchCore, Filterjam, Combustor and Blackhole Reverb. Musically, the performance is divided into 2 sections, each incorporating live samples in 2 distinct stylistic contexts. A brief pause and a verbal announcement serve as markers for the transition from ambient to rhythmic-based music.

In the final stage of the performance, the audience actively participates by voting for their preferred section. This stage not only empowers the audience to voice their preferences but also signifies the official closure of the participation cycle.

After the performance, informal discussions with the audience ensue, allowing the audience to articulate their experiences in a more relaxed setting. These informal discussions are highly valuable to the performer and often inform subsequent participatory performances, particularly concerning the themes that could be explored in the questions posed for live sampling.

3. Technical requirements

Setup and soundcheck time – 40 minutes. Playback format – stereo.

Equipment	**Use for**	**Input**	**Provided by**
iPad x 2 (with 1 stereo ¼" output)	Audio Mixer	DI box (stereo) x 2	performer
iPhone x 1 (as above)	Audio Mixer	DI box (stereo) x 1	performer
Microphone for announcements (Shure SM58)	Audio Mixer		venue
Back Line			
Stage Monitors			
Stereo foldback	performer		venue
Stage Hardware			
Audio Mixer (3 stereo line and 1 mic input)	equipment		venue
Table/space for equipment	equipment		venue

References

Borgo, D. (2022). “Sync or Swarm: Improvising Music in a Complex Age”. New York, Bloomsbury Academic.

Breel, A. (2015). “Audience agency in participatory performance: a methodology for examining aesthetic experience”, In: Participations, 12, p. 368−387.

Hashtag Lab (2023). “iubar project”. https://hashtaglab.pl/event/iubar/ Accessed 15 September 2023.

Koszolko, M K (2021). “Performative Storytelling: The Model of Setting-Based Mobile Music Creation”, In: Mobile Storytelling in an Age of Smartphones, p. 173−190. Edited by X Xu, and M Schleser. Cham, Palgrave Macmillan.

Koszolko, M K (2022). “Performative Storytelling: Setting-Based Mobile Music Creation in Action”, In: Proceedings of the Ubiquitous Music Symposium ubimus2022 (pp. 28−31). Edited by M Messina, D Keller, L Costalonga and F Ribeiro. G-ubimus.

Toelle, J, & Sloboda, J A. (2021). “The audience as artist? The audience’s experience of participatory music”, In: Musicae Scientiae, 25(1), p. 67−91.

Workshops

Workshop on Body Percussion and Live-Electronics

Nyokabi Kariũki[1] (KE), Alex Hofmann[2] (AT)

[1]Independent Artist – Nairobi, Kenya

kariuki.nyokabi@gmail.com

[2]Department of Music Acoustics (IWK) – University of Music and Performing Arts Vienna, Austria

hofmann-alex@mdw.ac.at

Abstract. *This workshop focuses on the artistic perspectives of Nyokabi Kariũki, a composer and performer from Nairobi, Kenya, who uses voice and body percussion to explore the complexities of post-colonial African identity, and Alex Hofmann, a live-electronics performer and researcher investigating questions of live-electronic performance preparation strategies. The workshop encourages participants to actively explore body percussion playing techniques and to develop new concepts for combining body percussion with live-electronics considering approaches from the field of ubiquitous music.*

1. Workshop Description

Within the umbrella of the Ubiquitous Music (Keller et al. 2023), specifically related to embodied and ecological approaches to music practices, during this workshop the topic of Body Percussion and Live-Electronics will be explored from multiple perspectives: First, from the artistic perspective of composer and performer Nyokabi Kariũki from Nairobi, Kenya, who is currently working on a longform experimental sound work that will serve as her exploration of the complexities of post-colonial African identity. In this work, she centers herself as a performer on voice and body percussion, the latter of which plays a vibrant role in various folk musical traditions across Africa. Second, from the perspective of live-electronics performer and researcher Alex Hofmann, who is investigating questions of live-electronic performance preparation strategies. Kariũki and Hofmann are currently exploring the intersections of their respective research. During the workshop, they will demonstrate some of the music created under this collaboration; where live-electronics are used in conversation with the body, exploring the way the body's sound can be understood, enhanced, and disrupted (see video link in Section 3).

Workshop participants will be invited to actively explore body percussion playing techniques during the workshop and are encouraged to develop and discuss new concepts for combining body percussion with live-electronics. This will concern questions such as: Which role can live-electronics play in a body percussion performance? How can a body percussion performer control the live-electronics? What types of DIY sensors and open-source software tools are applicable? Which restrictions need to be considered when combining body percussion with electronics? How to develop live-electronic performances for a larger ensemble of body percussion performers? Furthermore, the presenters would like to ignite some conceptual discussion about our relationship to our body as sound makers. Can using our body as a vessel of sound help us become more

attuned to ourselves? What can we learn from the African musical traditions from which these movements are borrowed?

2. CVs of Workshop Presenters

Nyokabi Kariũki is a Kenyan composer and sound artist based between Kenya and the United States. Her sonic imagination is ever-evolving, spanning across genres from classical contemporary and experimental-electronic music, to sound art, pop, film, and (East) African musical traditions. She performs with the piano, voice, electronics, and on several instruments from the African continent. Her concert works have been performed by notable ensembles such as Cello Octet Amsterdam and Third Coast Percussion, and she has received commissions from BBC Radio 3, Heartland Marimba, and more. Released in February 2022, her debut EP, 'peace places: kenyan memories' (SA Recordings) was described as "deft" (The Quietus) and "transcendent" (The Guardian), with Bandcamp highlighting seeing her as "becoming a crucial voice in contemporary composition and experimental music". Nyokabi seeks to create meaningful and challenging art, illuminated by a commitment to the preservation and reflection on African thought, language and stories. Website: https://www.nkariuki.com/

Alex Hofmann is a professor at the Department of Music Acoustics (IWK) at mdw – University of Music and Performing Arts Vienna, a sound artist and a saxophone / live-electronics performer. He investigates tools and methods to enhance expressiveness in music performance by advancing creative music technologies.

Website: https://iwk.mdw.ac.at/hofmann/

3. Related Video Material

"Experiments on Body Percussion and Live-Electronics (Helsinki 2023)", by Nyokabi Kariũki and Alex Hofmann: https://youtu.be/S1haP9iRZkY?si=rLnSb18-bvh9516f

"Questions on the Origins of Creativity," by Nyokabi Kariũki: https://www.youtube.com/watch?v=UOw59jlHqPo

4. Funding

This research was supported by the Austrian Science Fund (FWF): AR-743

References / Related Literature

Devenish, L. (2022). Instrumental infrastructure: sheet materials, gesture and musical performance. In Embodied Gestures. TU Wien.

Keller, D., Yaseen, A., Timoney, J., Chakraborty, S., and Lazzarini, V. (2023). Banging Interaction: A Ubimus-Design Strategy for the Musical Internet. Future Internet 15, no. 4: 125. https://doi.org/10.3390/fi15040125

Kiiru, K., & wa Mũtonya, M. (Eds.) (2018). Music and Dance in Eastern Africa: Current Research in Humanities and Social Sciences. Nairobi: Africae. doi:10.4000/books.africae.1622

Mitchell, T. J., Madgwick, S., & Heap, I. (2012). Musical interaction with hand posture and orientation: A toolbox of gestural control mechanisms.

Artistic Works

Signal to Noise Loops v5: Breathing Space - Reflections on Covid-19

Stephen Roddy (University College Cork)

Signal to Noise Loops v5. is a data-driven audio-visual installation for headphone presentation over mobile or smart devices. It is informed by principles from the fields of Cybernetics, Sonification, Generative Music, and Artificial Intelligence. Music and visuals were created using data from noise level monitors at Ranelagh, Bull Island, and Chancery Park. The first portion of the piece reflects data from April 2022, after the initial shock of the COVID-19 pandemic had subsided and human activity was returning to pre-pandemic levels in real world physical spaces across the city. The second reflects data from April 2020, at the height of the pandemic, as offline activity in real-world locations was tightly constrained and virtual or networked modes of communication became the dominant paradigm for human interaction.

Treatise- Sound Installation

Patrick Moore (Ulster University)

This idea for this piece is traced back to the words of Irish philosopher George Berkeley: 'But, say you, surely there is nothing easier than for me to imagine trees, for instance, in a park, or books existing in a closet, and nobody by to perceive them. The objects of sense exist only when they are perceived; the trees therefore are in the garden... no longer than while there is somebody by to perceive them.' (A Treatise Concerning the Principles of Human Knowledge, section 23, and 45- George Berkeley 1710)

These writings form the basis of the common adage: "When a tree falls in a lonely forest, and no animal is nearby to hear it, does it make a sound?" (1910 Physics- Charles Riborg Mann and George Ronsom Twiss)

When considering this as a source of inspiration for a contemporary music composition, the concept of perception of art as an interaction itself was considered; to use the audience's power to perceive the artwork then cause an effect on the outcome of the artwork itself. Interaction and co-creation between the audience and the installation, as well as interaction between multiple audience members is highlighted in this piece. This piece also showcases DIY Electronics in a creative capacity. This piece explores the concept of threshold of hearing and listening- in terms of volume, rhythm perception, and pitch. As audiences interact with the physical elements of the installation, the audio elements themselves evolve and fluctuate in tandem.

Ciúnas: antiphonal to am bisonic (8-channel)

Multichannel Audio/ Fixed-media concert
Declan Tuite (Dublin City University, Ireland)

Ciúnas is based around place, texture and resonance. The piece explores spatial composition, recording and mixing. Voices are recorded as single lines and individual notes/ vocalisations in the studio. The studio recorded voices source is played back into sites chosen for their acoustics properties and textures and thematic significance and the source re-recorded on site via ambisonic and stereo pair mics.

Four choral pieces composed specifically for this project, with spatial arrangement in mind, offer the basis of structure and narrative cohesion.

Ciúnas (Quiet - Stillness - Calm)
Leanbh - (Child)
Cá - (Where)
Ná - (No)

The voices produce no recognisable words but offer a wide range of textures, colour and tone which interact with the particular material properties of the spaces. (e.g., Connemara Marble Quarry, Red Sandstone of Freiburg Cathedral, or the dense timber of Rosscahill Woods). Through rearranging the source playback and combining various ambisonic recordings impossible spatial arrangements and textural combinations are achieved.

While the principal concerns are resonance and space a narrative is realised via the choice of location which relates to the overriding theme of each piece to offer a throughline in each piece. For example, Cá mixes recordings from a mine, a wood and cave, while Ciúnas brings together a church, a government building, and a Magdalene Laundry. The pieces will be realised via three principal formats which each allow for an intentional ceding to the visitor/audience of the final mix while engaging with the piece.

Mortal Coil (Stereo)

Shane Byrne (Technological University of the Shannon, Ireland)

Mortal Coil is a composition that makes use of the inaudible electromagnetic fields (EMFs) which surround our quotidian existence. This piece was initially conceived as a companion to a series of images that incorporated the near-infrared spectrum of light in their composition. In this capacity, both visual and sonic elements would expose the audience to a particular experience that is usually hidden from our senses.

Ideological Distortion (Stereo)

Berk Yagli (University of the Arts, London, UK)

Ideological Distortion is a piece which explores the dark side of today's media, dilution of ideologies, and constant bombardment of confusion. It invites the listener into reflecting on the issues and feel the horror and hate that is constantly imposed on society whether we individuals are lucid about it or not.

Timios Stavros (8-channel)

Iain McCurdy (Maynooth University, Ireland)

The sources for *Timios Stavros* were the result of a week's field recording in western Crete in April 2023. The sound survey aimed to gather not just sounds identified with that location, but also their refraction through Crete's unique geology – mountain tops, valleys, gorges - and man-made spaces of churches and alleyways. The title of this piece is taken from the name of a hilltop church where a number of recordings were made. It is a common name for such located churches and translates as 'holy cross'.

It is a soundscape composition which aims to heighten the degree of capture and immersiveness of these spaces through the use of surround-sound recording techniques and highest quality recording equipment. An ancillary aim of the composition was the development of a plugin for the decoding of double-mid-side (DMS) recordings into an arbitrary number of channels on any 2d plane location, expanding on the 5-speaker limit of the existing Schoeps plugin. The undecoded version of this piece is therefore a 3-channel sound file which can be decoded for any floor-level speaker configuration.