Opensemble:
A framework for collaborative algorithmic music

Matías Zabaljáuregui
Programa de Investigación: Sistemas Temporales y Síntesis
Espacial de Sonido en el Arte Sonoro.
Escuela Universitaria de Artes. UNQ
matias@ventenmusic.com

Diego Dorado
Venten Music
diego@ventenmusic.com

ABSTRACT

This article introduces Opensemble, a framework for collaborative algorithmic music which employs the software engineering techniques used by the Open Source development model to discover the music that this kind of distributed and large scale collaboration can offer.

This proposal combines the author's prior research with current aesthetic concerns and an interest in exploring recent ideas behind Open Innovation for computer music. This work presents the first motivations of the framework.

1. INTRODUCTION

Opensemble is a framework created to explore the possibilities that arise when composing algorithmic music following the Open Source development principles. This paper is the first public document of our work in progress.

The pieces are entirely written using Supercollider programming language, with the use of GitHub as a collaborative platform and a model based on *trusted developers*, similar to that generated in the kernel Linux community [1]. Evaluation of the contributions is based solely on their technical qualities and artistic relevance towards the work at hand. A model of meritocracy is followed; a form of government in which hierarchical positions are reached based solely on merit [2].

The challenges to be met are many: how are the contributions organized? How is the work divided? What software engineering strategies can be incorporated into algorithmic composition? What effects does public communication during the development of the work produce?

Motivations for Opensemble stem from diverse sources of research, such as exploration into computer music, network music and applying the Open Source model as a paradigm for collaborative composition. Found below, are the concepts arising from these three disciplines which will allow us to define and outline our proposal with greater precision. For lack of space we will not deal with the designs based on tactile collaborative interfaces [3, 4], nor will we analyze laptop orchestras. [5, 6].

2. A COLLABORATIVE APPROACH FOR ALGORITHMIC COMPOSITION

There are many works which deal with the current and historical advances in algorithmic composition and computer assisted composition. Among the most distinct is that of [7], which focuses on Artificial Intelligence techniques and [8], which provides an introduction to the basic techniques and complementary examples of creative work. Collins [9] attempts to explore the possibilities for musical form in algorithmic composition from a psychological perspective and Nierhaus [10] performs a review of a great variety of methods used for algorithmic composition, with further investigation being done in [11] where a dialectical work performed by 12 composers and the different algorithmic composition techniques used is presented.

This work assumes that algorithmic composition will become an ever more complex activity. The implementation of hybrid systems which combine several algorithmic approaches has led to new possibilities of expression. Nevertheless, "*The main disadvantage of hybrid systems is that they are usually complicated, especially in the case of tightly-coupled or fully integrated models. The implementation, verification and validation is also time consuming."* [12]. An example of increasing complexity is shown by Musical Metacreation (MuMe) which combines several disciplines to study the automatization of musical creativity. In [13] the need is discussed: "*...to explore collaborative methodologies in order to make meaningful creative and technical contributions in the field. With the release of the Musebot specification, such opportunities are possible through an open-source, community based approach…*".

2.1 Motivations behind Opensemble

A small number of publications mention some methodological problems in related research. It is therefore interesting, in the initial stages of our project, for these to be reviewed so as to avoid reproducing the errors that are mentioned.

In Fernández's article [7] phrases such as "*reinventing the wheel in algorithmic composition techniques was common*" and *"...artists, who tended to develop ad hoc solutions, and the communication with computer scientists was difficult...*" stand out. Opensemble proposes practices, tools and standard protocols imported from

Open Source which make communication easier and avoid the cost of having to create ad-hoc tools.

On the other hand, in Pearce's classic paper [14] , four possible high level motivations for writing software for musical composition are identified and the problems observed in literature as a result of being unable to distinguish the possible motivations, are later exposed. We have positioned our motivation among the first two mentioned by Pearce: in motivation 1, "*software written by the composer as an idiosyncratic extension to her own compositional processes"*, there are no methodological limitations and it is unnecessary to define rigorous criteria in order to define success, as the design of the software is part of the creative process. In motivation 2:"*software written as general tools to aid any composer in the composition of music"*, the problem becomes a software engineering task. Consequently, software engineering standards should be upheld.

Pearce encourages interdisciplinary work when there is more than a sole motivation, as is our case. Nevertheless, each motivation implies different methodologies and evaluation criteria. As a result, it will be necessary to take these overlapping motivations into consideration, in order to enrich interdisciplinary work and in turn define objectives and methods of evaluation for each.

An evident motivation in Opensemble relates to collaborative creation through distributed software engineering techniques. In this case "*Rather than exploring specific algorithms, this study focuses on system and component design."* [15]

Nevertheless, from the very beginning, our quest has been artistic/idiosyncratic. We consider that this work exposes a novel composition strategy, much as Open Innovation proposes new strategies of innovation in different fields [16, 17]. In the same way that algorithmic procedures for artistic creation were explored in the past, with no formal or theoretical objectives, we propose exploring modern software engineering techniques in order to discover the artistic results achievable. It is a *practical motivation*, as explained in [14].

3. OPEN SOURCE PARADIGM AS A FRAMEWORK FOR NETWORK MUSIC

There have been experiments performed with network music since the end of the 1970`s. Other artistic disciplines (*media art, net art and web art*) have indeed performed more thorough research in the past but it is our aim here to approach them from the perspective of algorithmic composition.

"*Designing and implementing a network music system supposes that new, meaningful sonic results can be achieved by collaboration over computer networks*" [18]. This is exactly the prime motivation behind our project. We can define Opensemble as a framework for implementing *interconnected musical networks*, as stated in [19]:" Making decisions about the motivations, social perspectives, and the network architectures are essential steps towards setting the framework for an effective musical network project."

Taking the exploratory work done in the field of network music into consideration, [18, 20, 21, 22], several "*popular conceptions of Internet music*", [23], can be observed. Here, special attention must be paid to '*Music that Uses the Internet to Enable Collaborative Composition or Performance*', as in the case of the FMOL Project [24].

Lancaster [25, 26] pose the question whether the current creation paradigm, more homogenous than the heterodox techniques used in the initial projects of this discipline, is an authentic musical need or a simple convenience. This transformation "*has brought with it aesthetic questions about the reason and evolution of this new genre.*" [26]. Although not oriented towards performance, Opensemble offers the maximum flexibility for collaborative composition allowing for synchronous as well as asynchronous approaches [18] and supporting both the sequential horizontal approaches, in which composers add consecutive fragments, as well as the vertical approach, which allows for overlapping of voices and sounds or modification of pre-existing material. [24]

3.1 Social Organization

It has been observed recently that "*Network music ensembles are uniquely positioned to deploy heterarchical technologies that enable them to address radical democratic concerns relating to communication structures and power distribution."* [27]. As it is later explained in this work, Opensemble is a clear example of this statement.

Two questions are central when classifying a project as Network Music:

- *What are the goals and motivations for designing a musical network?*
- *What are the social perspectives, architectures, and network topologies that can be used to address these goals and motivations?*

Because of its motivations, Opensemble is classified as a structure-based system (as opposed to process-based) "*In structure-based systems, the main goal of the interaction tends to focus on its outcome"* [19]. Composers and designers of such systems are usually more interested in aspects such as artistic vision and compositional arrangement instead of educational or social experience of members involved.

From the perspective of social organization, Opensemble mimics the kind of government used by the Linux development community: "*uses Peer Production methods and can be considered a Virtual Network Organization with a Peer Governance structure. [28]"*

Opensemble follows Open Source [29] institutional design and proposes *"a bazaar-like approach to coordination and leadership so as to allow a core team, trusted lieutenants, and other motivated contributors to emerge". "As a project matures, new key developers can emerge* *based on the concept of meritocracy [where] status is earned based on the merit of a developer's contributions" [2].*

Due to dependence on the work of volunteers, recruiting and holding on to the collaborators is a crucial factor for the success of a project of its kind. Consequently, it is necessary to understand the factors that affect the motivation of the developers [30, 31]. "*Furthermore, the threat of forking limits the ability of project leaders to discipline members."* [32]. That is most likely the reason why a *"review of literature identifies governance as an area of significant interest in the open source research community."*[33].

4. FIRST PROTOTYPE. DESIGN AND IMPLEMENTATION

The first prototype, named "*LHCVMM, Large Hadron Collider Visual Music Machine*" [34], is a visual music project based on data generated by the Large Hadron Collider [35], the world's largest and most powerful particle accelerator, located at CERN in Switzerland. The goal is to translate the data generated by the ATLAS detector [36], one of the four major LHC experiments, into stimuli for the musical composition and performance.

Since we've begun, our intention has been to solve problems as they arise based on the needs of each individual project. Soon enough several questions emerged:

- *What common language could we adopt to describe a piece of music without tying it to a specific work?*
- *How to translate ATLAS data into a descriptive piece of music to drive both music and visuals?*
- *What will be the framework to manage collaborations in such a work?*

4.1 Adoption of a common language

Rather than creating a new language to describe a piece of music, we adopted Denis Smalley's Spectromorphology [37] defined as the perceived sonic footprint of a sound spectrum as it manifests in time.

As proposed by Manuella Blackburn in her paper [38], we use spectromorphology to create a sort of musical score consisting of sound unit events over time. In data-based musical pieces, the data driving the music needs to be translated to generate this score. Otherwise, the score is created during the composition stage.

We defined a spectromorphological vocabulary as well as the sound unit data structure. Each sound unit may have three phases respectively called onset, continuant and termination. Each phase is comprised of several properties describing its growth and motion, spectrum, and texture motion.

4.2 Data translation and Score composition

Publicly available ATLAS datasets, are queried with the help of pyROOT, a python extension module that allows us to sift through this data with ease. Collission events are then translated into sound units based on their kinematic properties and other characteristics of the overall process. This collection of sound units represents the aforementioned score which is streamed as OSC messages.

This strategy allow us to write the score on a spreadsheet, utilize a simple script to read each row as a sound unit, and stream them as OSC messages. Although initialy meant to prototype LCHVMM, multiple scores can be written with these tools, proving the design to be functional across musical pieces.

4.3 Framework and Collaborations

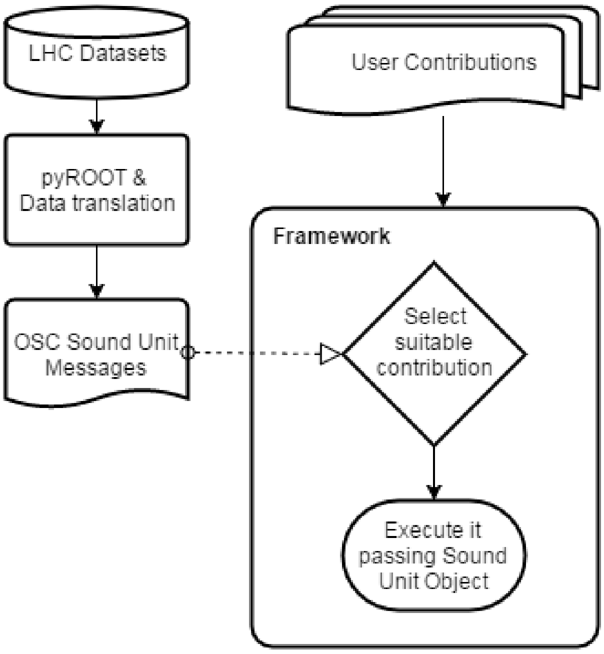
Having defined the vocabulary, the structure of sound units, the generation of scores and the mechanism to stream it through OSC, we must now exhibit how to turn this into playable collaborative music.

A framework was developed in Supercollider responsible for managing collaborations, receiving sound unit OSC messages, and finally reproducing the music. Collaborators register functions that can implement sonically sound units matching certain vocabulary terms by calling a method on the framework. Those functions receive a sound unit object as argument allowing collaborators to use its properties on their implementation. Finally, the framework, listening to OSC messages, selects the most suitable registered function for each received message and pass the sound unit object to the selected function. This selection is done based on the best match of vocabulary terms of function descriptions and sound unit object properties.

To recap, the framework performs the following actions:

1. All collaborations found in a special folder are registered.
2. An OSC listener is started to receive sound unit messages.
3. Upon message reception, a suitable function is selected.
4. The sound unit object is passed to this function to be performed.

Although we expect this design to change greatly as we progress, we are confident we are on the right track as it has already proven to be an efective approach.


LHC Datasets
pyROOT & Data translation
OSC Sound Unit Messages
User Contributions
Framework
Select suitable contribution
Execute it passing Sound Unit Object

5. CONCLUSIONS

There is no doubt that Opensemble presents a variety of compositional and technical challenges. The algorithmic composer will have to incorporate new practices and tools, will have to be open to discussing his ideas in public mailing lists and will have to accept that his work be published under open licenses.

We believe this to be a natural consequence and an inevitable convergence of the aforementioned areas of investigation. Moreover, it represents a motivation to reassess some of the loopholes in current computer music research. In this regard, a lack of articles in the area of software engineering applied to collaborative algorithmic composition stands out. It is particularly interesting to study the possible applications of *Distributed Agile Development* (DAD) which has received increasing interest both in industry and academia [39].

"*Characterized by a globally distributed developer force, a rapid, reliable software development process and a diversity of tools to support distributed collaborative development, effective FLOSS (Free/Libre and Open Source Software) development teams somehow profit from the advantages and overcome the challenges of distributed work"* [40]

One of the main challenges of our work is to discover ways in which to incorporate the advantages of Open Source to specific projects of collective musical creation based in the Internet. It is essential then to point out the need for an interdisciplinary perspective. Opensemble is an experiment that in turn encloses network music, algorithmic composition and Open Source software engineering.

The experiences with Opensemble in these first months have been encouraging. The collaboration has brought about interesting designs and we are persuaded that there is much interesting music to be found.

6. REFERENCES

[1] C. Schweik. "The institutional design of open source programming: Implications for addressing complex public policy and management problems", *First Monday* [Online], 8.1, 2003.

[2]. P. Ågerfalk, B. Fitzgerald, and K. Stol, "*Software Sourcing in the Age of Open: Leveraging the Unknown Workforce"*. Springer, 2015.

[3] N. Klügel, M. R. Frieß, G. Groh, and F. Echtler, "An approach to collaborative music composition", *Proceedings of the international conference on new interfaces for musical expression,* 2011, pp. 32-35.

[4] N. Klügel, A. Lindström, and G. Groh, "A genetic algorithm approach to collaborative music creation on a multi-touch table", *40th International Computer Music Conference, ICMC 2014, Joint with the 11th Sound and Music Computing Conference, SMC 2014,* 2014, pp. 286-292.

[5] D. Trueman, et al, "PLOrk: the Princeton laptop orchestra, year 1", *Proceedings of the international computer music conference,* 2006, pp. 443-450.

[6] G. Wang, et al, "Stanford laptop orchestra (slork)", *Ann Arbor, MI: Michigan Publishing, University of Michigan Library*, 2009.

[7] J. D. Fernández and F. Vico, "AI methods in algorithmic composition: A comprehensive survey", *Journal of Artificial Intelligence Research* (2013): 513-582.

[8] J. M. Peck, "Explorations in algorithmic composition: Systems of composition and examination of several original works", *Master's thesis, State University of New York, College at Oswego*, 2011.

[9] N. Collins, "Musical form and algorithmic composition", *Contemporary Music Review*, vol. 28, no. 1, pp. 103–114, 2009.

[10] G. Nierhaus, *Algorithmic Composition: Paradigms of Automated Music Generation*, Springer Berlin / Heidelberg, 2009.

[11] G. Nierhaus, *Patterns of Intuition - Musical Creativity in the Light of Algorithmic Composition,* ISBN: 978-94-017-9560-9 (Print) 978-94-017-9561-6 (Online), Springer, 2015.

[12] G.Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects", *In Proceedings of the Symposium on Musical Creativity*, pp. 110–117, 1999.

[13] A. Eigenfeldt, O. Bown, and B. Carey, "Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble," *Proceedings of the Sixth*

International Conference on Computational Creativity, 2015

[14] M. Pearce, D.Meredith, and G. Wiggins, “Motivations and methodologies for automation of the compositional process”, *Musicæ Scientiæ,* vol. 6, no. 2, pp. 119–147, 2002.

[15] C. Ariza, “An Open Design for Computer-Aided Algorithmic Music Composition: athenaCL”, Ph.D. thesis, New York University, 2005.

[16] C. Herstatt and D. Ehls, *Open Source Innovation: The Phenomenon, Participant's Behaviour, Business Implications (Routledge Studies in Innovation, Organization and Technology),* 2015.

[17] J. Faludi, “Open innovation in the performing arts. Examples from contemporary dance and theatre production”, *Corvinus journal of sociology and social policy*, vol. 6, no. 1, pp. 47-70, 2015

[18] A. Barbosa, "Displaced soundscapes: A survey of network systems for music and sonic art creation." *Leonardo Music Journal,* vol. 13, pp. 53-59, (2003):

[19] G. Weinberg, "Interconnected musical networks: Toward a theoretical framework." *Computer Music Journal,* vol. 29, no. 2, 2005, pp. 23-39.

[20] M. Akkermann, “Computer Network Music Approximation to a far-scattered history”, *Proceedings of the Electroacoustic Music Studies Network Conference - Electroacoustic Music Beyond Performance*, Berlin, 2014.

[21] M. Ayers, *Cybersounds: Essays on virtual music culture*, Peter Lang, 2006.

[22] W. Duckworth, *Virtual music: How the Web got wired for sound*. Routledge, 2005.

[23] A. Hugill, "Internet music: An introduction.", *Contemporary Music Review*, vol. 24, no. 6, 2005, pp. 429-437.

[24] S. Jordà, M. Alonso, and Grup de Tecnología Musical. "MÚSICA E INTERNET, CREACIÓN, INTERCAMBIO Y EDUCACIÓN", 2006.

[25] S. Gresham-Lancaster, “Computer Network Music”, *Arts, humanities and complex networks*. Sci 2013. Retrieved from: http://ahcncompanion.info/abstract/computer-network-music.

[26] S. Gresham-Lancaster, “Computer Music Network”, *Leonardo*, vol. 47, no.3, pp. 266-267, 2014.

[27] S. Knotts, “Changing Music’s Constitution: Network Music and Radical Democratization”, *Leonardo Music Journal*, vol. *25*, pp. 47-52, 2015.

[28] Linux Governance. Retrieved from: http://p2pfoundation.net/Linux_-_Governance

[29] E. S. Raymond, *The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary*, O'Reilly Media, Inc, 2001.

[30] D. Ehls, C. Herstatt, “Open Source Participation Behavior-A Review and Introduction of a Participation Lifecycle Model”, *In 35th DRUID Celebration Conference*, 2013.

[31] G. Von Krogh, S. Haefliger, S. Spaeth, and M. W. Wallin, *Carrots and rainbows: Motivation and social practice in open source software development*, *MIS quarterly*, vol. *36*, *no.* 2, pp. 649-676, 2012.

[32] Y. Li, C. H. Tan, and H. H. Teo, “Leadership characteristics and developers’ motivation in open source software development”, *Information & Management*, vol. *49, no.* 5, pp. 257-267, 2012.

[33] A. Blekh, “Governance and organizational sponsorship as success factors in free/libre and open source software development: An empirical investigation using structural equation modeling. Doctoral dissertation”. Nova Southeastern University. Retrieved from NSUWorks, Graduate School of Computer and Information Sciences. http://nsuworks.nova.edu/gscis_etd/40. 2015.

[34] Large Hadron Collider Visual Music Machine. Available at: https://github.com/Opensemble/lhcvmm/

[35] The Large Hadron Collider. Available at: http://home.cern/topics/large-hadron-collider

[36] About the ATLAS Experiment. Available at: http://atlas.cern/discover/about

[37] D. Smalley, “Spectro-morphology and Structuring Processes”, *EMMERSON, S. The Language of Electroacoustic Music London, Macmillan Press Ltd*, 1986, pp. 61-93.

[38] M. Blackburn, "Composing from spectromorphological vocabulary: proposed application, pedagogy and metadata" , *Novars Research Centre, The University of Manchester.*
Retrieved from : http://www.ems-network.org/ems09/papers/blackburn.pdf

[39] M. Paasivaara, S. Durasiewicz and C. Lassenius. "Using scrum in distributed agile development: A multiple case study." *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on* 13 Jul. 2009: 195-204.

[40] K. Crowston, K. Wei, J. Howison and A. Wiggins, “Free/Libre open-source software development: What we know and what we do not know”, *ACM Computing Surveys (CSUR),* vol. *44, no.* 2, pp. 7, 2012.