

Conservatoire National Supérieur de Musique et de Danse de Paris

Année scolaire 2012-2013

2^e année de 2^e cycle supérieur

**La musique de Tristan Murail
et
la composition assistée par ordinateur,**

Autour de la musique mixte par l'analyse de *Pour adoucir le cours du temps*

Travail d'étude personnel

Keita MATSUMIYA

Tuteur : Mikhail MALT

Mai 2013

Même signant ce travail tout seul, il est cependant un travail d'équipe. Plusieurs personnes et entités que je cite n'auront pas conscience, ou idée de l'importance qu'ils ont eue dans le développement de cette étude, néanmoins, j'aimerais les citer comme signe de ma reconnaissance envers leur participation explicite ou implicite.

Au compositeur Tristan Murail, pour m'avoir permis d'accéder à ses documents informatiques.

Au professeur Laurent Pottier, pour m'avoir fait confiance et pour avoir partagé ses archives.

Au tuteur Mikhail Malt, pour responsable par l'aventure qu'a consisté dans l'écriture de ce mémoire.

A professeur Gérard Pesson, pour ses conseils et ses encouragements.

A la professeure Anne Robet, m'avoir fait confiance et m'avoir donné l'occasion de développer ce travail.

A l'Ircam, pour avoir été un lieu de multiples rencontres et possibilités.

Et finalement, à Hiromi.

Table des matières

Introduction.....	3
I Contexte de <i>Pour Adoucir le Cours du temps</i> de Tristan Murail	
I-1 Travail entre compositeur et informatique musicale.....	4
I-2 Commande de GMEM.....	5
II Introduction de composition assistée par ordinateur	
II-1 OpenMusic et les méthodes liées.....	6
II-2 Pratique de programmation dans l'OpenMusic.....	7
III Processus de générer des objets musicaux de <i>Pour Adoucir le Cours du temps</i>	
III-1 Structure de temps.....	11
III-2 Structure harmonique.....	24
III-3 Analyse de section.....	27
IV Conclusion	43
ANNEXE I.....	44
ANNEXE II	51
ANNEXE III	53
ANNEXE IV.....	56
ANNEXE V.....	57
BIBLIOGRAPHIE.....	58

Introduction

Mon sujet se veut une réflexion sur *Pour adoucir le cours du temps* (2005) pour ensemble et électronique de Tristan Murail qui est un compositeur consacré sa réflexion compositionnelles autour de l'harmonie et du timbre de point de vue de spectre sonore. Je porterai sur cette pièce afin de dévoiler ses pensées musicales de ce moment par rapport à ses utilisations de nouvelles technologies.

La motivation de cette étude provient d'abord de mes intérêts sur l'informatique musicale en tant que compositeur. En suite, les dernières années, on trouve de nombreux travaux qui décrivent la nouvelle technologie ou des outils pour la composition, mais il existe peu d'écrits décrivant comment le compositeur s'est servi de ces technologies. Dans cet article, j'essaie d'examiner le rapport entre le résultat – la partition, l'enregistrement de Murail – et les traces de pensée – la programmation du compositeur –.

La méthodologie de recherche sera en principe des couches de vérifications successives autour de recherches technologiques avec l'écriture musicale de Tristan Murail par rapport à la composition assistée par ordinateur (CAO) qui ont permis d'approfondir sa pensée.

Première étape, j'envisagerai son concept autour d'une harmonie issue du spectre sonore, soit d'une "harmonie spectrale" qui s'effectue sur son système d'harmonie, de figures, de courbe mélodique dans son écriture, avec la vérification de son patch avec un logiciel OpenMusic qui lui permet de générer son écriture par le moyen d'analyse et de la resynthèses.

Tristan Murail reste toujours un des plus importants pionniers de la composition assistée par ordinateur, et il serait nécessaire de vérifier dans cette pièce ses découvertes musicales et sa nouvelle écriture musicale qui soit particulièrement attachée autour de la potentialité des nouvelles technologies de cette époque.

Finalement, je m'appliquerai à la considération de ses recherches dans cette pièce afin de mettre aux séries d'œuvres pour ensemble et électronique dans la trajectoire de ce compositeur, car ce sont des effectifs d'ensemble de 11 à 18 instruments ou de 5 à 6 instruments qui lui ont surtout fait expérimenter avec les dispositifs électroniques et les nouveaux logiciels de chaque période.

I Contexte de *Pour Adoucir le Cours du temps* de Tristan Murail

I-1 Travail entre Compositeur et Informatique Musicale

Avant la composition de *Pour adoucir le cours du temps*, Murail avait considéré, au moins depuis 20 ans, d'adopter la potentialité de sciences et les nouvelles technologies pour son besoin musical. A travers ses activités de l'ensemble L'Itinéraire entre les années 70 et 80, avec ses camarades de compositeurs, y compris Michael Levinas, Gérard Grisey, et Hugues Dufour, il avait préparé de développer ses idées autour d'un nouveau monde sonore comportant l'électronique.

Sa musique était dans un premier temps composée sur une continuité : celle de matériau sonore. Ensuite, l'idée de «processus» qui correspond à une modification progressive d'un état sonore pour en obtenir un second va imprégner ses œuvres. Un processus qui peut se réduire à un son ou à un geste peut enchaîner d'autres processus. Les processus permettent également de concilier les contraires, d'absorber, d'intégrer. Afin de traiter plus quantitativement différents paramètres de la musique, Murail est plus approché du monde de la science avec l'aide de la puissance des nouvelles technologies. Les œuvres mixtes pour ensemble et électronique, y compris le «Désintégration» (1982) et «L'Esprit des dunes» produites à l'Ircam (1994) sont des résultats remarquables de telles recherches.

Dans la production de ses travaux, la collaboration avec des développeurs et des scientifiques est essentielle. Il s'est consacré aussi pendant des années au développement de l'outil de composition pour réaliser ses idées. En effet, Murail a été nommé comme premier professeur du Cours de Composition, fondé à l'Ircam dans le département pédagogique en 1991. Le cursus a pour but de donner aux jeunes compositeurs des occasions d'apprendre l'utilisation des outils et de nouvelles pensées autour de l'informatique musicale par rapport à ses activités créatives. L'Ircam ont ouvert ses portes dans le secteur d'éducation, de faire augmenter des chances de collaborer pour des compositeurs avec l'informatique musicale dans l'environnement de nouvelles outils. Il a quitté sa poste en 1997 pour aller aux Etats-Unis à enseigner à l'Université Colombia. Actuellement, il travaille au Mozarteum à Salzbourg en Autriche.

Laurent Pottier, qui a collaboré avec Murail en tant que réalisateur en informatique musicale dans le projet de «Pour adoucir Le Cours du Temps», il avait suivi le cours de doctorat au cursus de musicologie dirigé par Hugues Dufour à l'Ircam, après avoir obtenu une maîtrise de sciences à l'université. A partir de ses intérêts sur l'expression musicale avec des outils de nouvelles technologies, il a décidé de travailler au Cours de Composition à l'Ircam en 1993. Quand Murail a

quitté l'Ircam en 1997, il a quitté l'Ircam en même année pour continuer ses recherches au GMEM, au Centre national de création musicale de Marseille¹.

I-2 Commande de GMEM

En 2004, le GMEM a commandé une œuvre mixte à Murail, de sorte que Murail et Pottier ont été réunis sur ce projet, du fait qu'ils avaient déjà travaillé ensemble à l'Ircam. Dans la production de la composition, Murail avait utilisé depuis de nombreuses années un langage de programmation développée en 1989 à l'Ircam, le Patchwork². Le Patchwork avait été utilisé par de nombreux compositeurs, y compris Kaija Saariaho, Jean-Baptiste Barrière et Murail à partir de début des années 90s. Murail était aussi assidu à développer des bibliothèques spécifiques du Patchwork pour étendre ses fonctionnalités à son goût, en collaboration avec des équipes de l'Ircam, ont également été compris Laurent Pottier à l'un des ingénieurs. En 1998, Patchwork a évolué comme l'OpenMusic par l'équipe de Carlos Agon et ses collègues de l'Institut. Lors de la collaboration de Murail et Pottier au GMEM entre 2004 et 2005, les fonctions nécessaires de patchwork ont été mise à jour pour s'adapter à la bibliothèque de l'OpenMusic, qui est utilisé dans le projet de « Pour adoucir Le Cours du Temps ».

Après avoir quitté l'Ircam en 1997, Murail avait travaillé en tant que professeur à l'Université Columbia jusqu'en 2012. Quand il a eu une commande par GMEM à 2004, il avait cherché la potentialité d'utiliser dans son œuvre le logiciel Spear, développé par Michael Klingbeil, l'un de ses élèves à l'Université. Ainsi, Laurent Pottier a lancé de développer une bibliothèque de sorte qu'il puisse utiliser des données de Spear dans l'OpenMusic.

A cette époque au GMEM, Pottier était en train de développer des applications autour de l'analyse-synthèse sonore par modélisation physique. Ils ont été conçus pour être utilisés comme générateur des figures musicales détectées à partir de l'objet sonore comme expiration, bruit ou eau. Pottier a tenté d'appliquer ces recherches en cours à la collaboration avec Murail. Il a lancé le projet de développer des bibliothèques, afin d'être accessible dans l'OpenMusic les données obtenues par ses applications de GMEM et par le Spear, et puis il s'est consacré à préparer des programmes pour la réalisation de partie électronique par le jeu de clavier MIDI au concert. Le travail a duré un an et demi³.

¹ Entretien avec Laurent Pottier, janvier 2013.

² Le logiciel, Patchwork, a été conçu et développé par Michael Lauson à l'Ircam en 1989.

³ Entretien avec Laurent Pottier, janvier 2013.

II Introduction de Composition Assistée par Ordinateur

II-1 OpenMusic et les méthodes liées

L'OpenMusic, conçu et développé par Gérard Assayag, Carlos Agon, et Jean Bresson à l'Ircam dans les années 90, est un langage de programmation visuel basé sur CommonLisp / CLOS⁴. Un programme visuel est créé en ramassant et reliant des icônes représentant les fonctions et les structures de données. La plupart des programmes et opérations sont effectués en faisant glisser une icône dans un endroit particulier et le déposer à un autre endroit. Les structures de contrôle visuel (par exemple, les boucles) sont intégrées dans ce programme, qui s'interface avec ceux de Lisp.

L'OpenMusic peut être utilisé pour les buts généraux de programmation fonctionnelle, de celle d'objective et de celle de visuelle. À un niveau plus spécialisé, un ensemble de classes et les bibliothèques fournies font un environnement pratique pour la composition musicale. Le projet OpenMusic vit au-dessus du noyau OpenMusic. Il est un ensemble spécialisé de classes et de méthodes écrites en Lisp, accessibles et visualisables dans l'environnement OpenMusic. Les différentes classes qui implémentent les données et les comportements en musique sont fournies. Ils sont associés à des éditeurs graphiques et peuvent être prolongés par l'utilisateur pour répondre aux besoins spécifiques. Les différentes représentations d'un processus musical sont traitées, y compris la notation commune, le piano-roll MIDI, le signal sonore. Une organisation de la matière musicale est proposée à un haut niveau dans le domaine de temps à travers le concept de maquette.

II-1 Pratique de programmation dans l'OpenMusic

Dans ce chapitre, nous allons vérifier la base de fonction de l'OpenMusic en voyant l'exemple de programmation.

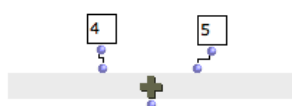


Figure 1 : Programme de «4 + 5» sur le patch

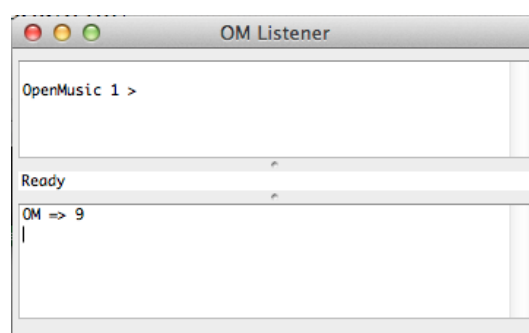


Figure 2 : Résultat sur la fenêtre de OM Listener

⁴ Le Common Lisp Object System, souvent abrégé en CLOS (prononcé 'si-lauss'), est l'ensemble des primitives présentes dans le langage de programmation Common Lisp pour construire un programme orienté objet.

Il s'agit d'un processus simple avec 3 objets «4» et «5», et «OM+». L'objet «OM+» représente la fonction d'addition. Cette programmation représente le calcul de l'équation «4 + 5 = 9»⁵. Quand on clique le point bas de «OM+» en maintenant la touche commande sur Mac, on obtient le résultat dans la fenêtre «OMListner». Dans tous les objets de l'OpenMusic, il y a des points qui représentent l'entrée ou la sortie de l'information. Et la programmation se fait par la connexion de lignes entre des objets. L'espace de programmation est appelé un patch. Quand on clique la sortie de quelque objet, le calcul lance et on peut confirmer le résultat dans la fenêtre «OM Listener» comme image au-dessus.

Ensuite, nous allons voir le calcul avec des notes de musiques.

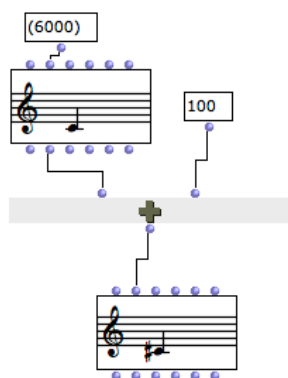


Figure 3 : Calcul avec des notes sur le patch

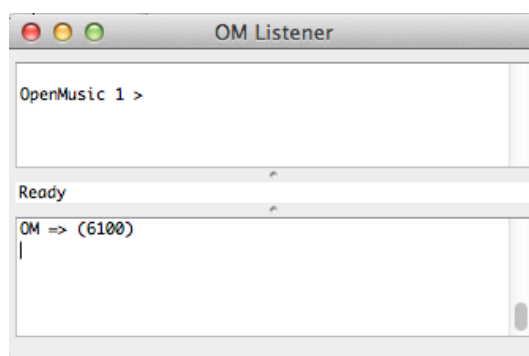


Figure 4 : Résultat sur la fenêtre de OM Listener

Le *Do*3 mis sur Clé de *sol* est représenté par le nombre 60 selon la règle internationale de MIDI⁶. Cette boîte numérique est appelée en anglais «Note Number» dans la règle. Le *Do*# 3, demi-ton en haut de *Do* 60 est la «Note Number» 61. En OpenMusic, pour le traitement des micros intervalles, on utilise une unité «MIDI Cent» qui est créée par la multiplication de 100 sur la valeur de «Note Number». L'exemple ci-dessus, le nombre 6000 entre sur l'objet «Chord» qui visualise la hauteur sur la portée traditionnelle, et puis l'addition de 100 sur 6000 est réalisée sur la fonction «OM+» et le résultat renvoie sur l'objet «Chord». Si on fait une évaluation de l'objet «Chord», le résultat 6100 est indiqué dans la fenêtre OMListner.

L'exemple suivant traite de la notion de liste.

⁵ L'addition de 4 et 5 est représenté en Lisp la suivante : (+ 4 5).

⁶ «Musical Instrument Digital Interface» est un protocole de communication et de commande permettant l'échange de données entre instruments de musique électronique, un ou plusieurs de ces « instruments » pouvant être des ordinateurs. Il est géré par l'International MIDI Association.

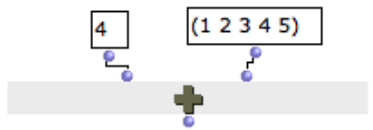


Figure 5 : Programme de liste sur le patch

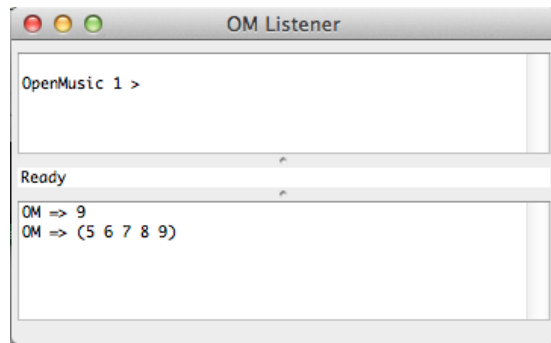


Figure 6 : Résultat sur la fenêtre de OM Listener

Les nombres mis entre parenthèses sont également utilisés sur la fonction connectée, et on dit cet enchaînement une liste comme unité des informations. Ici, la fonction «OM+» calcule l'addition de 4 pour chaque nombre mis entre parenthèses comme «4+1 4+2 4+3 4+4 4+5», et il renvoie le résultat sur le point bas de l'objet. On voit le résultat (5 6 7 8 9) dans la fenêtre droite.

Alors, nous allons voir le calcul qui traite la liste de notes de musique.

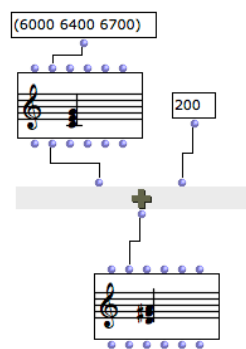


Figure 7 : Calcul de liste avec des notes sur le patch

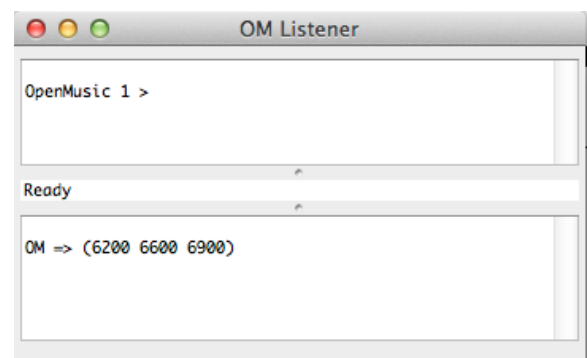


Figure 8 : Résultat sur la fenêtre de OM Listener

Une liste de nombres (6000 6400 6700), sont envoyés à la fonction «Chord» et on voit l'accord *Do Mi Sol*. Et puis, l'addition de 200 sur chaque nombre est effectuée sur «OM+» et le résultat est indiqué au «Chord» en bas comme *Ré Fa La*. On voit le résultat d'une «List» (6200 6600 6900) dans «OMListener».

Quand des plusieurs parenthèses se trouvent à de différents niveaux en profondeur, on peut traiter une sous-liste comme objet indépendant.

$$((a\ b\ c\ d)\ (1\ 2\ 3\ 4)\ (x\ y\ w\ z)) \Rightarrow (\alpha\ \beta\ \gamma)$$

Nous allons vérifier cette fonction avec l'exemple simple de programmation «List» dans «List».

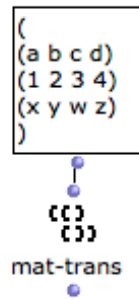


Figure 9 : Fonction de remplacement des objets

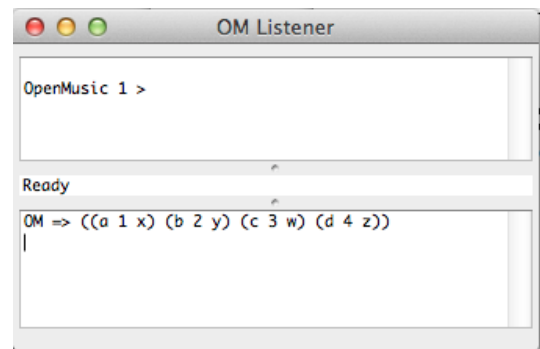


Figure 10 : Résultat de remplacement

La fonction «mat-trans» numérote d'abord chaque objet de sous-liste pour identifier sa position, et puis elle regroupe des objets selon ses positions avant de remplacer ces objets dans «List».

Voici le processus ;

((a b c d) (1 2 3 4) (x y w z))
=> Position : ((1^{er} 2^{me} 3^{me} 4^{me}) (1^{er} 2^{me} 3^{me} 4^{me}) (1^{er} 2^{me} 3^{me} 4^{me}))
=> Remplacement : ((1^{er} 1^{er} 1^{er}) (2^{me} 2^{me} 2^{me}) (3^{me} 3^{me} 3^{me}) (4^{me} 4^{me} 4^{me}))
=> Résultat : ((a 1 x) (b 2 y) (c 2 w) (d 4 z))

Ensuite, nous allons voir le calcul qui traite la «List» dans «List» avec des notes de musique.

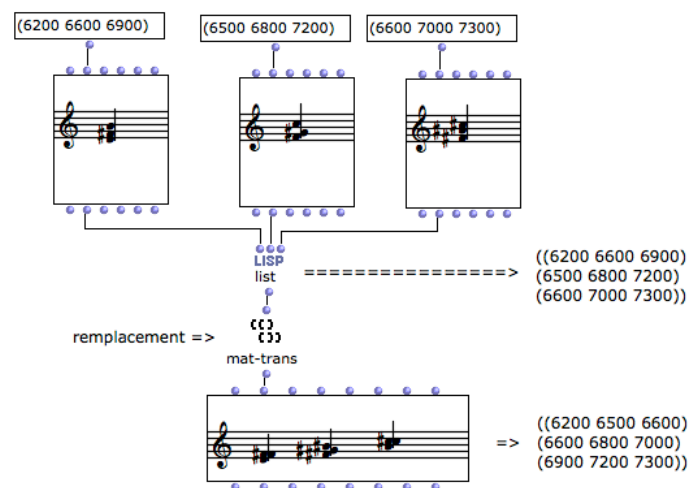


Figure 11 : Remplacement des notes et son résultat de remplacement

3 accords ont été rassemblés par la fonction «LISP list», et puis ces informations sont remplacées par «mat-trans». La fonction la plus basse est «Chord-seq» qui peut traiter plusieurs accords ou des figures mélodiques, c'est-à-dire l'information de «List» dans «List».

En outre, sur certaines fonctions comme «Chord» ou «Chord-Seq», on peut vérifier en détail leurs contenus sur une autre fenêtre par double clique sur l'Objet.

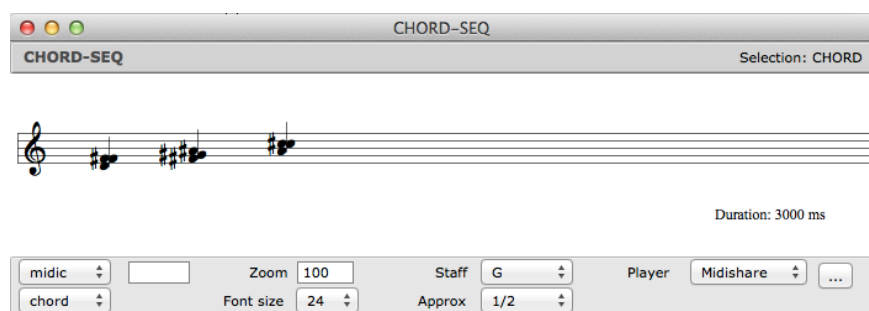


Figure 12 : Fenêtre de contenus de «chord-seq»

Comme les exemples expliqués au-dessus, nous pouvons traiter des notes de musique à travers des données numériques. En combinant des informations comme lettres, accords, séquences des notes, des graphes, durées, des équations complexes avec de divers paramètres, on peut construire des idées constructives pour figurer des productions créatives. En outre, en utilisant des bibliothèques qui élargissent des domaines à traiter, on peut faire entrer certaines données d'audio, des informations extérieures et les faire sortir. Effectivement, Murail utilise plusieurs sortes de bibliothèques originales dans sa programmation, comme «OMTristan», «OMspdata», «OM2Csound», «OM_LP_TM», ainsi de suite. La plupart de ces bibliothèques dérivent de la production par les réalisateurs en informatiques musicales à travers des discussions directes avec le compositeur. Ces fonctions n'ont pas simplement le calcul, mais ce sont des fruits de l'idée à construire des phénomènes autour de la musique.

Dans le chapitre suivant, nous allons vérifier ses idées principales avec des exemples de programmation que Murail a effectivement utilisés dans sa partition.

III Processus de générer des objets musicaux de *Pour Adoucir le Cours du temps*

Afin de réaliser ses idées musicales dans son œuvre, Tristan Murail utilise la programmation de l'OpenMusic dans divers domaines comme la génération d'accords, génération de rythmes, ou la construction du temps. Dans ce chapitre, nous allons vérifier ses idées caractéristiques avec la construction des objets.

III-1 Structure harmonique

Plusieurs catégories d'objets musicaux vont et viennent, se transforment, s'entrelacent: une énorme aspiration (colorée de la résonance filtrée d'un gong) - les résonances colorées du même gong - un contour mélodique microtonal - une mélodie d'accords (ou de timbres, c'est comme on veut, ou comme on l'entend) - divers sons bruités, soufflés, granuleux, mais harmonisés de l'intérieur - une cloche à vache alpine, comme le tic-tac irrégulier d'une horloge relativiste.⁷

D'une manière générale, les processus musicaux de Murail sont basés sur l'enchaînement des matériaux harmoniques qui ont été générés par certaines logiques différentes. Quant aux générations des accords, il prépare plusieurs sortes de palette afin de construire son propre processus musical avec chaque matériel.

Voyez ci-dessous Figure X, se trouve l'un des principes de générer des accords qui ont servi à la base de cette œuvre. C'est un exemple de programmation par l'échelle harmonique naturelle basé sur le fondamental de 4800 de *Do*, et du 1^{er} au 16^{me} composants spectraux.

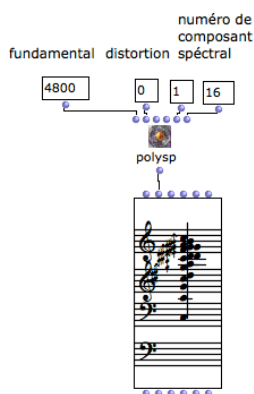


Figure 13 : Fonction «polysp» de «OMTristan»

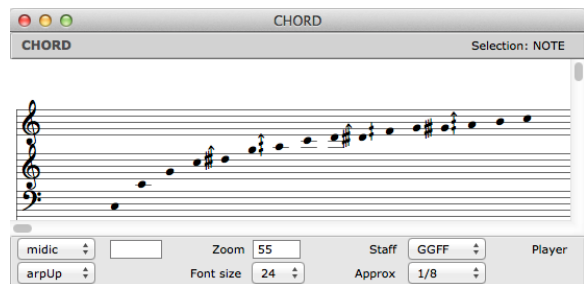


Figure 14 : «Contenu de Chord» par la visualisation d'échelle

L'image droite est la visualisation des hauteurs sous forme d'arpège sur l'objet «Chord». Si on modifie le taux de distorsion à 0.77, on obtient une autre échelle modulée comme ci-dessous.

⁷ Note de programme, Murail 2005.



Figure 15 : Changement de taux

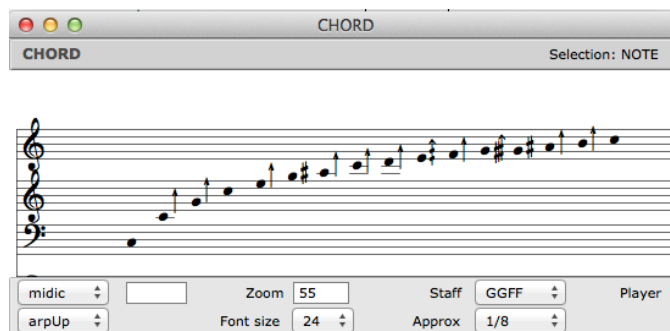


Figure 16 : Echelle obtenu par la distorsion

Ainsi, cette base de programmation a la potentialité de générer des variations d'échelle avec la manipulation à la manière de dilatation et compression de ses profils.

La suivante est la programmation accomplie par le compositeur afin de préparer des matériaux de «Pour adoucir le cours du temps», basés sur la fonction «polysp» qui était examinée avec le dernier exemple.

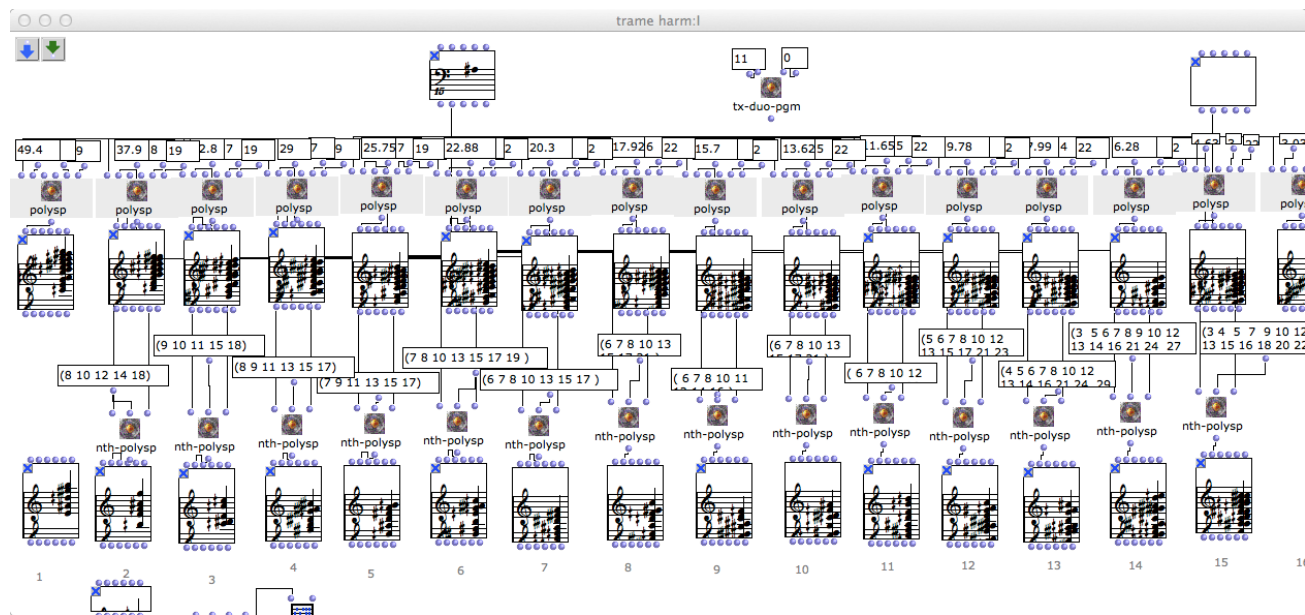


Figure 17 : Tableau des matériaux d'harmonie par sections 1

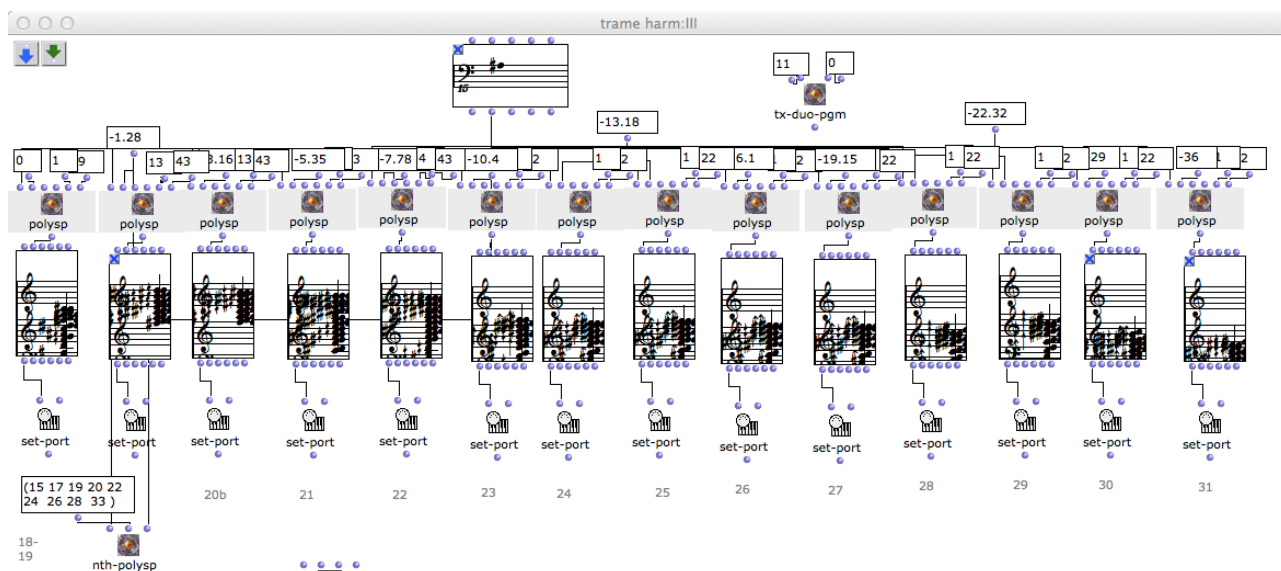


Figure 18 : Tableau des matériaux d'harmonie par sections 2

Ce sont des images du plan total, autour de conduit des accords au travers les 31 sections de cette œuvre. Tous les accords sont générés par le fondamental *La* de 4800 MIDI Cent, avec des variations de nombres des composants spectraux et des taux de distorsion. On y constate une cohérence et une diversité à travers des transitions d'accords. Murail utilise ces accords déconstruits par notes dans chaque section, combinés avec d'autres accords créés par objets sonores, en allant et en venant entre eux.

Ensuite, nous allons voir la programmation autour de l'utilisation des enregistrements de sons, y compris gong et alpenglocken, et sa manière d'appliquer tels matériaux. Au-dessous, on voit l'image des ondes de la sonorité de gong joué par une arche de cordes.

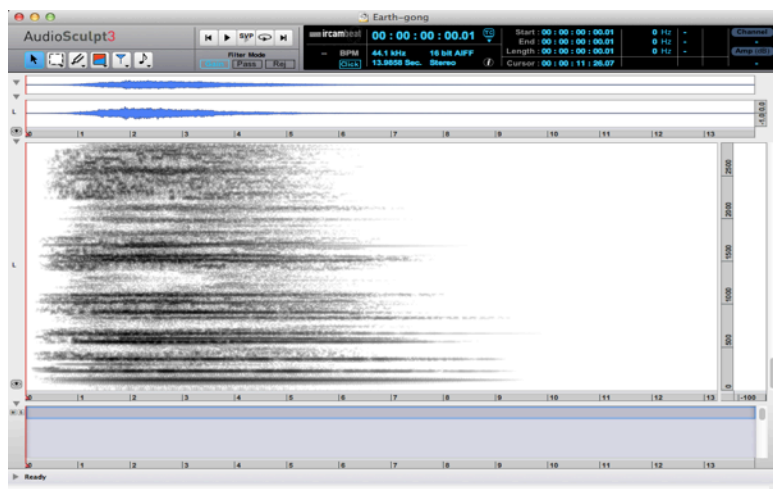


Figure 19 : Spectrogramme de la progression de temps par AudioSculpt ⁸

⁸ AudioSculpt est un logiciel pour la visualisation, l'analyse et la transformation des sons. AudioSculpt offre plusieurs représentations

Pour faire intégrer cette sonorité dans ses matériaux d'harmonie, Murail a utilisé un logiciel «Spear»⁹ développé par son élève de l'Université Colombia. Avec cet outil, il a pu analyser des distributions des composants spectrales et l'énergie d'amplitude de chaque haute fréquence du son comme ci-dessous.

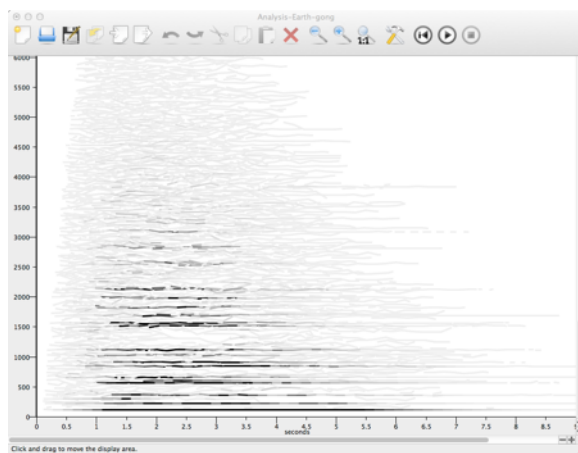


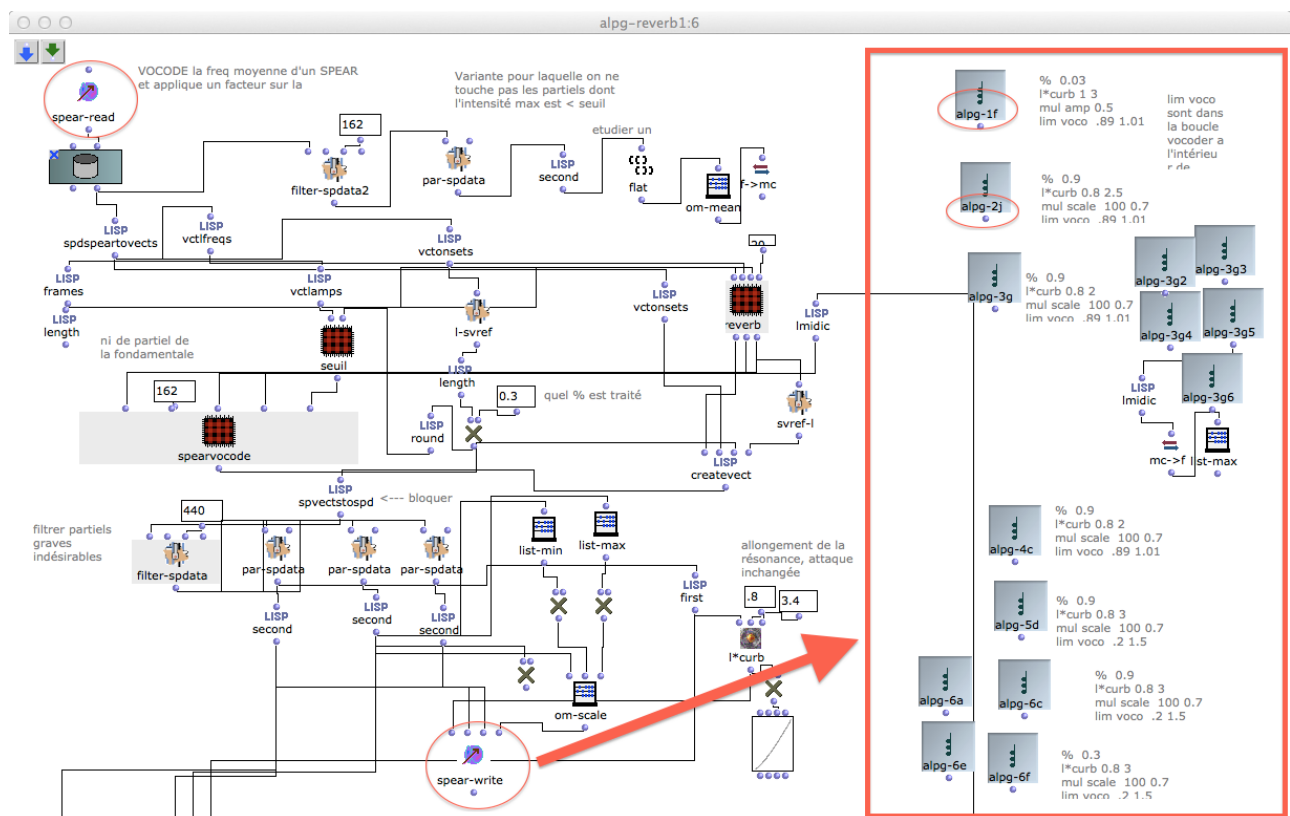
Figure 20 : Visualisation des partiels par Spear

Il s'agit d'une image d'un suivi de partiels du dernier son avec l'axe représentant le temps et l'axe Y représentant les. La densité de couleur de chaque bande représente l'amplitude de haute fréquence et la couleur indique sa force. La suivante est la programmation de Murail afin d'intégrer ces informations dans l'OpenMusic.

graphiques du son - forme d'onde, spectrogramme et sonagramme – pour obtenir les transformations souhaitées et aider l'utilisateur à choisir les paramètres de contrôle optimaux.

⁹ Un logiciel de la domaine de synthèse additive, SDIF, synthèse d'IFFT, suivi de partiels de LP, développé par Michael Klingbeil.

The image shows a musical score for two parts: 'Synthèse' and 'Clavier midi'. The 'Synthèse' part is written on a grand staff (treble and bass clefs). A red circle highlights the first note, 'la', in the treble clef. A large, shaded, diamond-shaped area covers the middle of the 'Synthèse' staff, indicating a sustained or complex sound. The 'Clavier midi' part is written on a single bass clef staff. The time signature is 4/4, indicated by a '4' and a vertical bar line. The tempo is marked 'Andante'.



1. *U. parvulus* (100%)
2. *U. parvulus* (100%)
3. *U. parvulus* (100%)
4. *U. parvulus* (100%)
5. *U. parvulus* (100%)
6. *U. parvulus* (100%)
7. *U. parvulus* (100%)
8. *U. parvulus* (100%)
9. *U. parvulus* (100%)
10. *U. parvulus* (100%)
11. *U. parvulus* (100%)
12. *U. parvulus* (100%)
13. *U. parvulus* (100%)
14. *U. parvulus* (100%)
15. *U. parvulus* (100%)
16. *U. parvulus* (100%)
17. *U. parvulus* (100%)
18. *U. parvulus* (100%)
19. *U. parvulus* (100%)
20. *U. parvulus* (100%)
21. *U. parvulus* (100%)
22. *U. parvulus* (100%)
23. *U. parvulus* (100%)
24. *U. parvulus* (100%)
25. *U. parvulus* (100%)
26. *U. parvulus* (100%)
27. *U. parvulus* (100%)
28. *U. parvulus* (100%)
29. *U. parvulus* (100%)
30. *U. parvulus* (100%)
31. *U. parvulus* (100%)
32. *U. parvulus* (100%)
33. *U. parvulus* (100%)
34. *U. parvulus* (100%)
35. *U. parvulus* (100%)
36. *U. parvulus* (100%)
37. *U. parvulus* (100%)
38. *U. parvulus* (100%)
39. *U. parvulus* (100%)
40. *U. parvulus* (100%)
41. *U. parvulus* (100%)
42. *U. parvulus* (100%)
43. *U. parvulus* (100%)
44. *U. parvulus* (100%)
45. *U. parvulus* (100%)
46. *U. parvulus* (100%)
47. *U. parvulus* (100%)
48. *U. parvulus* (100%)
49. *U. parvulus* (100%)
50. *U. parvulus* (100%)
51. *U. parvulus* (100%)
52. *U. parvulus* (100%)
53. *U. parvulus* (100%)
54. *U. parvulus* (100%)
55. *U. parvulus* (100%)
56. *U. parvulus* (100%)
57. *U. parvulus* (100%)
58. *U. parvulus* (100%)
59. *U. parvulus* (100%)
60. *U. parvulus* (100%)
61. *U. parvulus* (100%)
62. *U. parvulus* (100%)
63. *U. parvulus* (100%)
64. *U. parvulus* (100%)
65. *U. parvulus* (100%)
66. *U. parvulus* (100%)
67. *U. parvulus* (100%)
68. *U. parvulus* (100%)
69. *U. parvulus* (100%)
70. *U. parvulus* (100%)
71. *U. parvulus* (100%)
72. *U. parvulus* (100%)
73. *U. parvulus* (100%)
74. *U. parvulus* (100%)
75. *U. parvulus* (100%)
76. *U. parvulus* (100%)
77. *U. parvulus* (100%)
78. *U. parvulus* (100%)
79. *U. parvulus* (100%)
80. *U. parvulus* (100%)
81. *U. parvulus* (100%)
82. *U. parvulus* (100%)
83. *U. parvulus* (100%)
84. *U. parvulus* (100%)
85. *U. parvulus* (100%)
86. *U. parvulus* (100%)
87. *U. parvulus* (100%)
88. *U. parvulus* (100%)
89. *U. parvulus* (100%)
90. *U. parvulus* (100%)
91. *U. parvulus* (100%)
92. *U. parvulus* (100%)
93. *U. parvulus* (100%)
94. *U. parvulus* (100%)
95. *U. parvulus* (100%)
96. *U. parvulus* (100%)
97. *U. parvulus* (100%)
98. *U. parvulus* (100%)
99. *U. parvulus* (100%)
100. *U. parvulus* (100%)

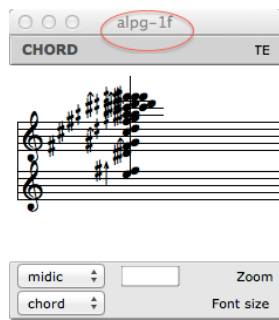


Figure 26 : Contenu de «alpg-1f»

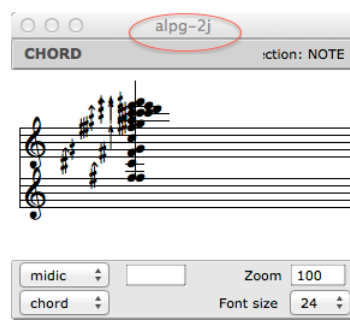


Figure 27 : Contenu de «alpg-2j»

Sur la partition, ils sont réalisés dans la partition par notation graphique comme ci-dessous.

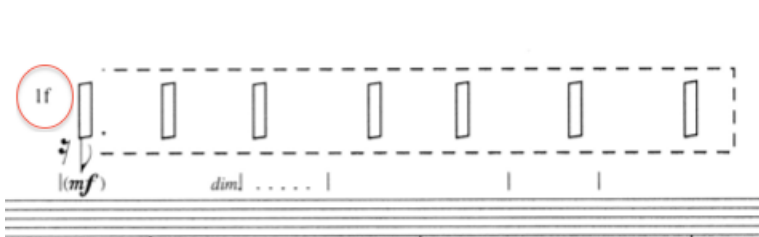


Figure 28 : Mesure 14, partie de synthèse

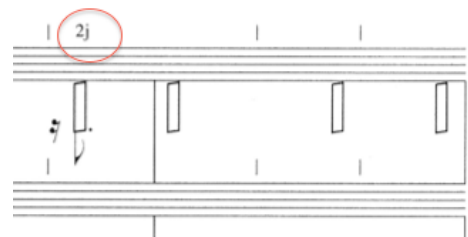


Figure 29 : Mesure 37, partie de synthèse

Parallèlement à la table des matériaux harmoniques comme ci-dessus, ils tiennent aussi des fonctions qui forment des figures mélodiques ou rythmiques.

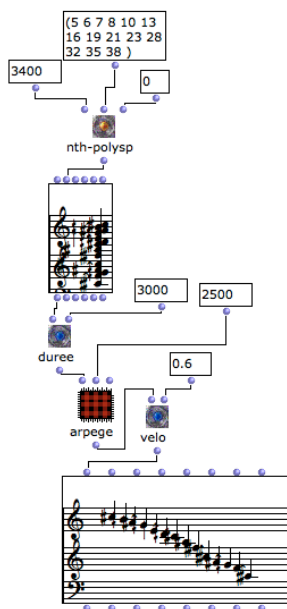


Figure 30 : Procédé de créer un arpegge

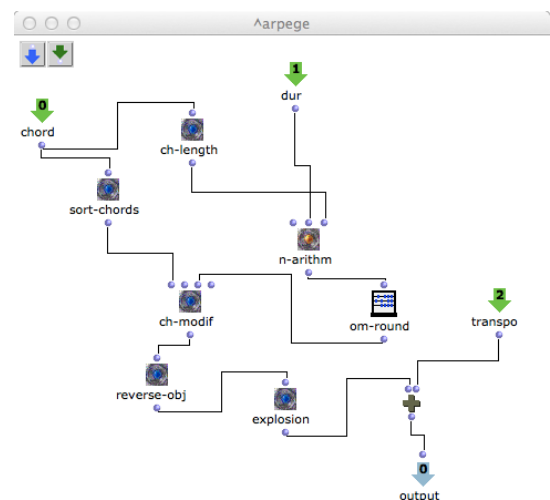


Figure 31 : Sub-patch, contenu de «arpege» de patch gauche

Après de la génération d'un accord, il ajuste les valeurs de notes sur l'objet «duree», puis à travers de sub-patch «arpege» qui rend varié le point de démarrage à chaque note, et à la fin, il modifie la dynamique des notes avec «velo».

Ensuite, c'est la transformation du temps sur le dernier arpège.

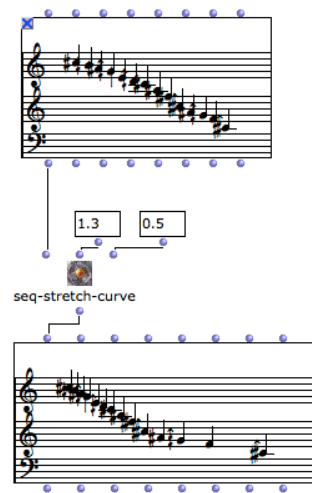


Figure 32 : Changement de valeurs des notes

En effectuant sur les valeurs de chaque note, l'objet «seq-stretch-curve» donne la modulation de proportion des lignes mélodiques dans le temps. Les matériaux obtenus ont été réalisés sur la partition comme ci-dessous. Pour toutes les données de notes dans «Chord» ou «Chord-seq», on peut varier le taux d'approximation, et ici, Murail indique le demi-ton d'approximation pour le piano.



Figure 33 : Mesure 30, partie de piano

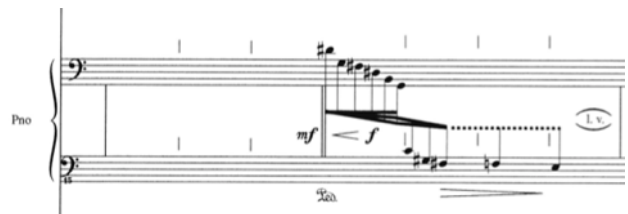


Figure 34 : Mesure 63, partie de piano

Ensuite, nous allons voir un patch qu'il génère les figures répétitives combinées des différentes progressions par l'accélération et le ralentissement, crée par même objet l'objet «seq-stretch-curve».

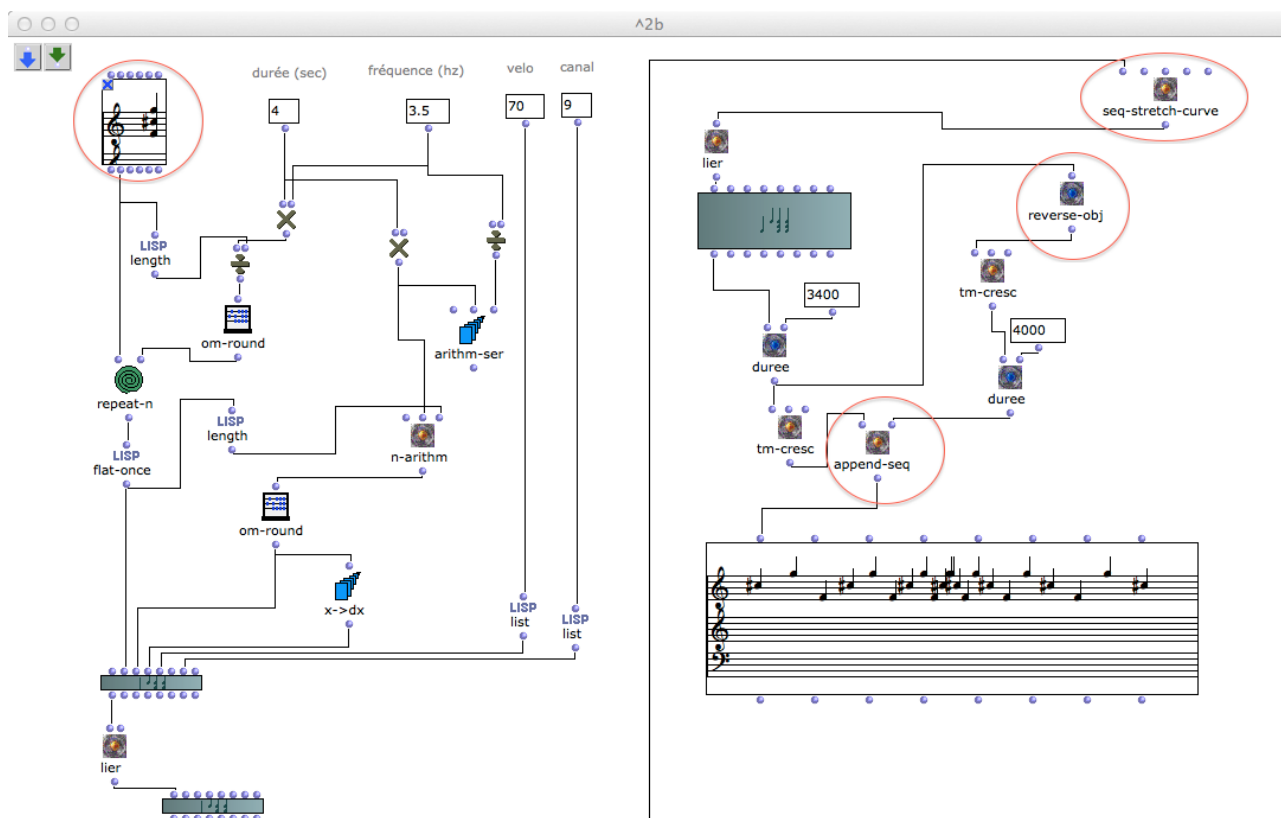


Figure 35 : Création de figures répétitives

Par l'accord en haut, ce patch crée d'abord une séquence d'accélération par l'objet «seq-stretch-curve», et puis l'objet «reverse-obj» génère une séquence inversée, et à la fin l'objet «append-seq» combine ces deux séquences.

Voici la réalisation de cette séquence dans la partition.



Figure 36 : Mesure 28, partie de flûte 1

Ensuite, il y a une autre manière de créer des séquences qui s'appelle «appel» dans sa programmation. Ci-dessous, on trouve des esquisses à créer des séries de mélodies de ce compositeur.

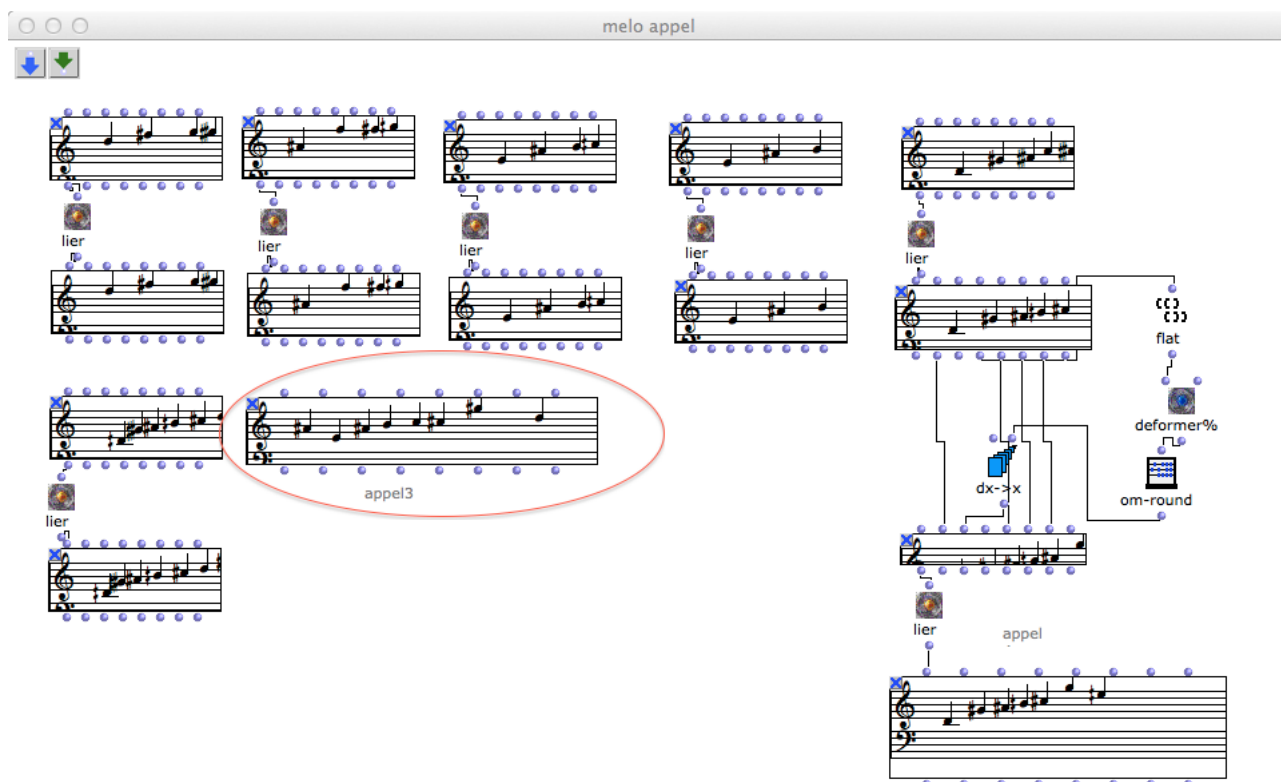


Figure 37 : Esquisse des mélodies «appel»

Ces mélodies ont été utilisées comme matériaux et réalisées aux plusieurs sections. Par exemple, «Appel3» de cette esquisse devient le matériel de section 3 et réalisé par flûte 2 comme ci-dessous.



Figure 38 : Mesure 58, partie de flûte 2

On trouve encore sa trace de pensée autour de ces séries de mélodies avec l'exemple suivant, le patch pour 4^e section. Dans le patch, il essaie de créer une polyphonie par ces données, et à la fin de ce patch, ces mélodies s'intègrent dans l'objet «multi-seq». Cet objet a la fonction de maintenir plusieurs «Chord-seq» dedans.

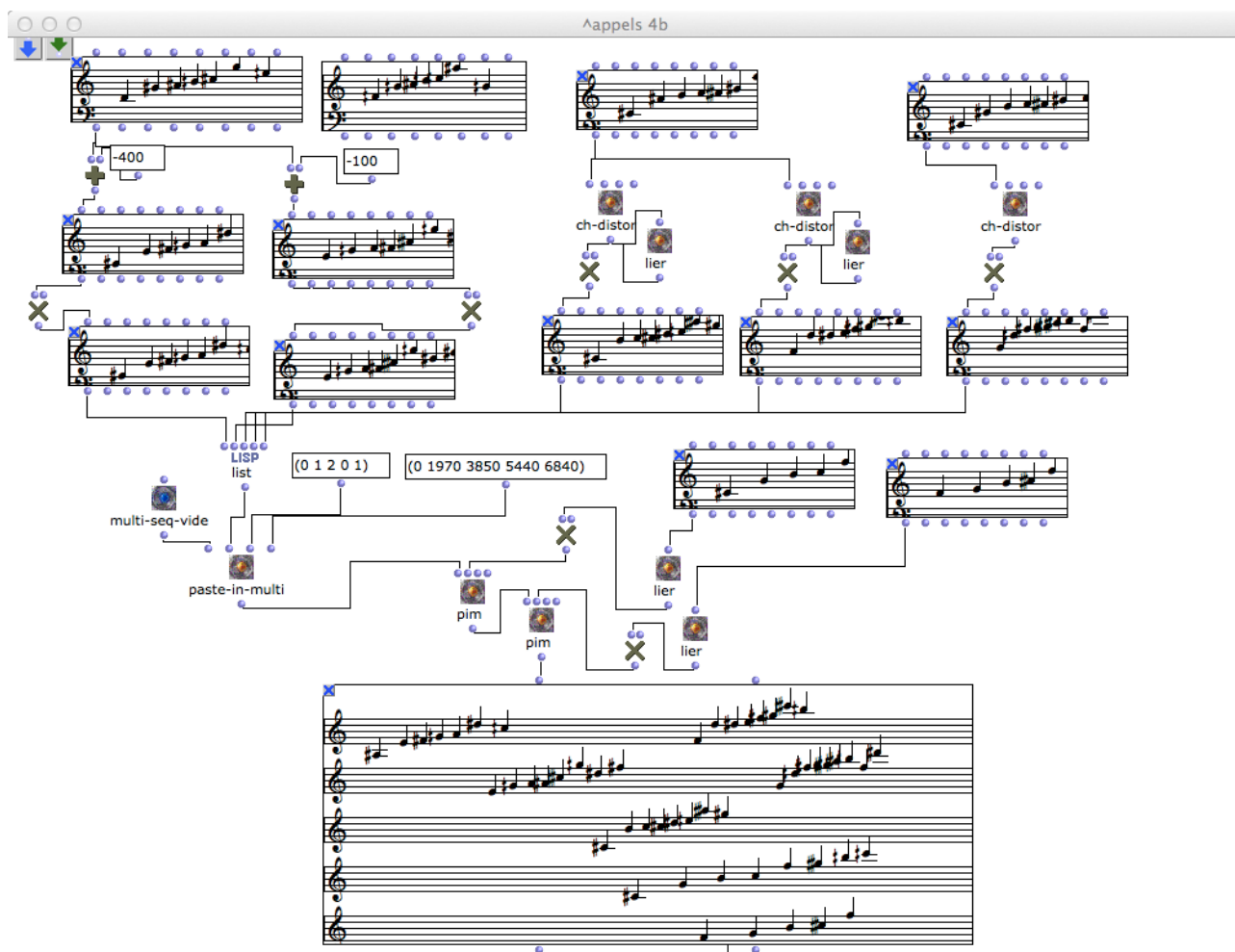


Figure 39 : Polyphonie de «appel»

Cette polyphonie va se développer par la suite avec la reconstruction de l'harmonie. Nous allons voir comment il a reconstruit l'harmonie pour ces échanges figuratifs dans sa programmation dans la figure suivante.

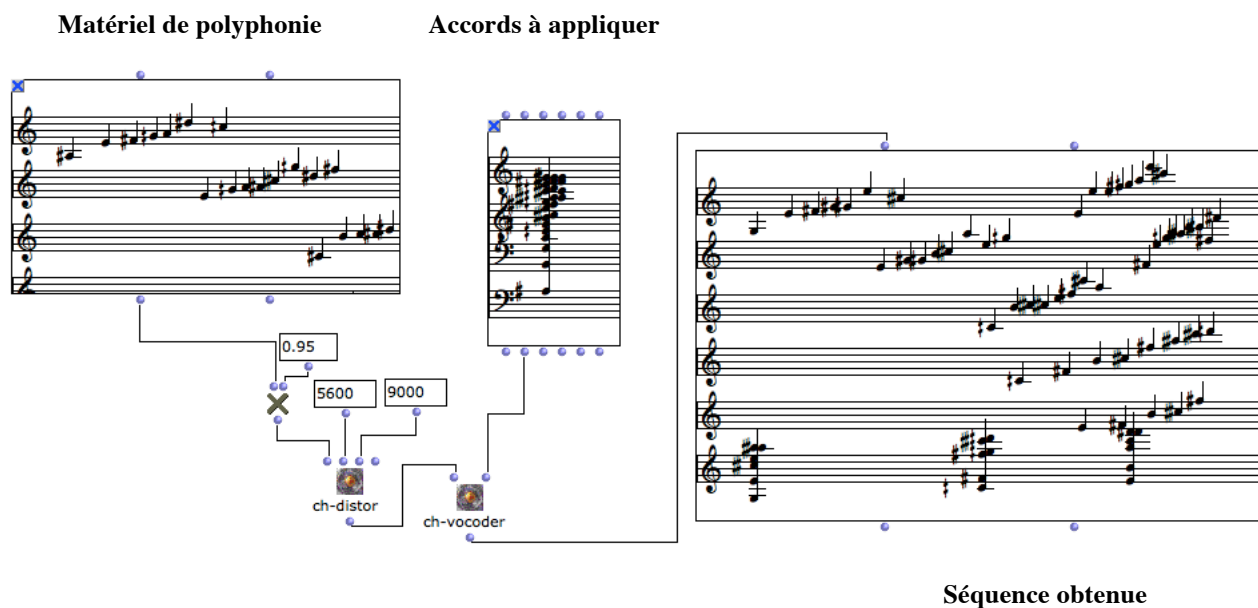


Figure 40 : Procédé de remplacement de l'harmonie

Avec la fonction «ch-vocoder», il remplace l'harmonie du matériel originel avec un autre accord préparé, en gardant des profils de chaque figure. Voici la réalisation sur la partition de cette séquence.

3 37

The image shows a musical score for the woodwind section of a piece, specifically measures 161 to 163. The score is written for six instruments: Flûte 1, Flûte 2, hautbois, clarinette 1, Clarinette 2, and Clarinette 3. The music is characterized by complex rhythmic patterns, including triplets and quintuplets, and dynamic markings such as *ff* (fortissimo) and *f* (forte). The notation includes various accidentals and articulation marks, indicating a highly technical and expressive passage.

Figure 41 : Mesure 161, partie de bois

Ainsi, il avait de diverses idées musicales sur la génération d'harmonie, sur le profil mélodique ou sur sa distribution aux voix avec sa programmation. En obtenant ces sortes d'objets musicaux, il les a mises dans le temps en utilisant la fonction de maquette.

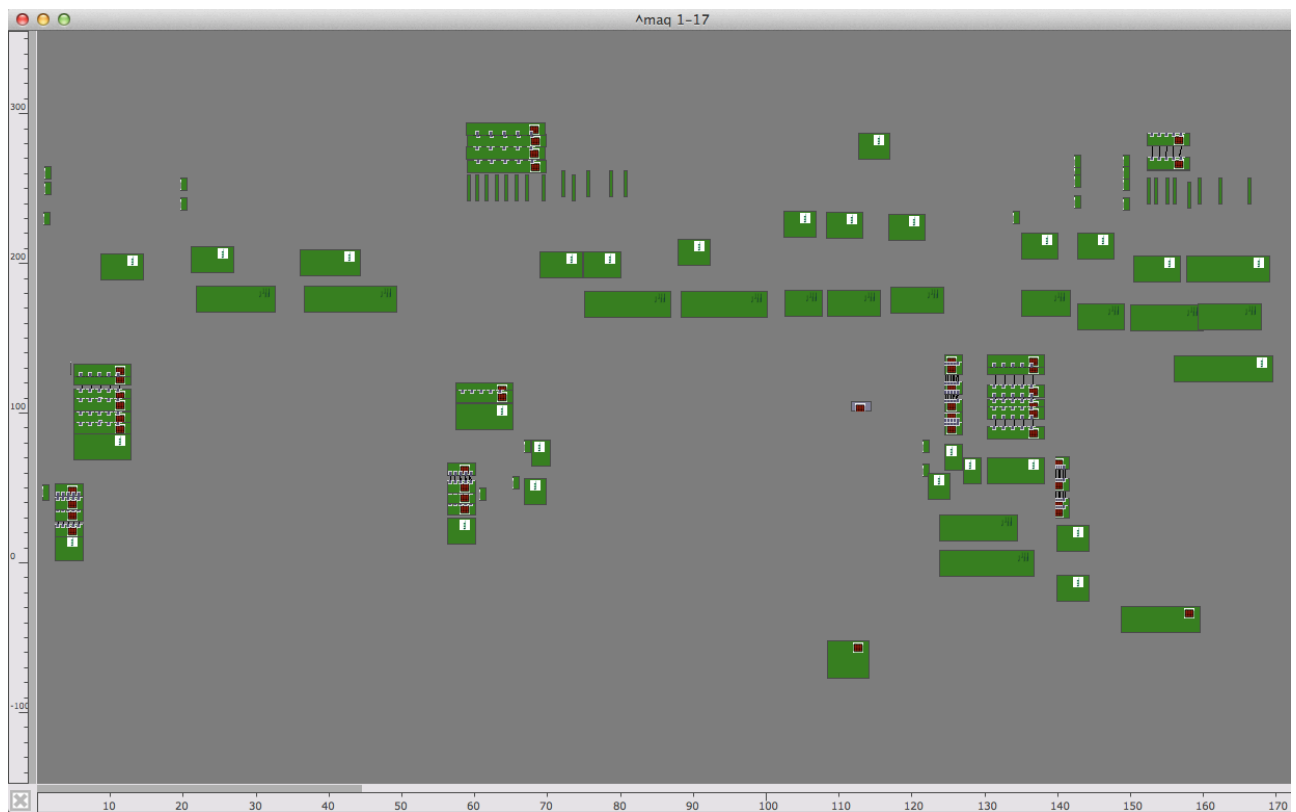


Figure 42 : Une maquette, extrait de début de la pièce *Pour Adoucir le Cours du Temps*

L'image ci-dessus est la visualisation de maquette. On y trouve plusieurs rectangles qui représentent des séquences comme «Chord-seq» ou «Chord». Grâce à cette maquette, mettre des objets musicaux sur la palette est égale à construire la progression des événements dans le temps. La maquette fait office de séquenceur de musique électroacoustique, et on peut modifier des sons en l'écouter. Les données de maquette sont transcrites à la main sur la partition instrumentale en ajoutant l'orchestration des figures.

Dans le chapitre suivant, nous allons voir la construction de temps en voyant sa programmation.