



**HAL**  
open science

# Espaces compositionnels et temps multiples : de la relation forme/matériau

Julia Blondeau

► **To cite this version:**

Julia Blondeau. Espaces compositionnels et temps multiples : de la relation forme/matériau. Acoustique [physics.class-ph]. Université Pierre et Marie Curie - Paris VI, 2017. Français. NNT : 2017PA066292 . tel-01717249

**HAL Id: tel-01717249**

**<https://theses.hal.science/tel-01717249>**

Submitted on 26 Feb 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# THÈSE DE DOCTORAT DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité  
**Composition musicale**

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par :

**Julia BLONDEAU**

Pour obtenir le grade de  
**DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE**

Sujet de la thèse :

## ESPACES COMPOSITIONNELS ET TEMPS MULTIPLES : DE LA RELATION FORME/MATÉRIAU

Soutenue le 13 décembre 2017

Devant le jury composé de :

M. Jean-Louis GIAVITTO  
M. Dominique PRADELLE  
M. Gérard BERRY  
M. Alain POIRIER  
M. Marco STROPPA  
Mme. Brigitte d'ANDREA-NOVEL  
M. Arshia CONT  
M. Philippe MANOURY

Directeur de thèse  
Encadrant de thèse  
Rapporteur  
Rapporteur  
Rapporteur  
Examinatrice  
Examineur  
Examineur

CNRS - Ircam  
Université Paris-Sorbonne  
Collège de France  
CNSMD de Lyon  
HMDK Stuttgart  
École des Mines  
Antescofo  
Collège de France

Julia Blondeau : Espaces compositionnels et temps multiples :  
de la relation forme/matériau, © Présentée et soutenue publi-  
quement le 13 décembre 2017.

# RÉSUMÉ

La technologie offre aujourd’hui aux compositeurs des outils et des langages permettant d’envisager de nouvelles approches et pratiques musicales, tant du point de vue de la conception du matériau que de celui de son articulation à la forme. Dans la musique mixte, l’avènement du « temps réel » entraîne avec lui de nouvelles possibilités d’écriture du temps et un rapport particulier aux processus qui le constituent.

Si les questions du temps, du matériau, de la forme et de leurs relations sont des questions centrales — et depuis bien longtemps — pour les compositeurs, il importe de se demander si la vieille dichotomie entre forme et matériau n’est pas à repenser en regard des nouveaux dispositifs susceptibles d’amener une appréhension à la fois plus fine et plus globale des diverses dimensions temporelles. Dans un tel contexte, et dans le cas plus précis de la musique mixte, la question que nous étudions est la suivante : dans quelle mesure l’existence d’un continuum entre les notions de forme et de matériau est-elle liée à la composition consciente d’entrelacs temporels ?

Cette question implique l’utilisation de paradigmes capables de nous faire appréhender finement les nombreuses composantes qui font l’identité d’un matériau ou d’une forme, et de nous faire percevoir les liens qu’elles entretiennent mutuellement. Elle implique également un langage permettant de parler de façon expressive des différents temps avec lesquels nous composons. Notre travail s’articule autour de deux axes complémentaires. Le premier traite de la nécessité d’organiser des espaces compositionnels ne se limitant pas à la production plus ou moins volumineuse d’un matériau mais pouvant intégrer, dans leur structure interne, des éléments d’échelles supérieures permettant, sinon de faire disparaître, au moins de rendre plus poreuse la frontière érigée entre les notions de forme et de matériau. Nous étudions à cet effet une représentation spatiale du matériau basée sur des structures topologiques. Le second nous amène à voir comment les derniers développements technologiques liés au langage synchrone Antescofo peuvent nous permettre de concevoir une musique mixte singularisée par les divers types d’écriture qu’ils sous-entendent et les nouveaux paradigmes qu’ils sont susceptibles d’engendrer. La composition de temps multiples, sur diverses échelles, liés par l’écriture et par des processus temporels propres aux paradigmes du temps réel, ainsi que par la prise en considération des temps liés à la performance, est abordée sous un angle résolument relationnel.

Cette recherche aborde donc des environnements compositionnels permettant d’agir sur plusieurs niveaux, de modéliser des processus hétérogènes intervenant à des échelles de temps différentes et d’élaborer une mise en relation entre elles. La recherche, intimement liée au travail de composition, s’établit à la fois dans le domaine de l’aide à la composition, au sens large, et dans celui de la composition dite du « temps-réel » à travers des outils et langages développés récemment à l’Ircam.



# REMERCIEMENTS...

... À l'ensemble du jury de cette thèse, aux rapporteurs qui ont très gentiment accepté de prendre sur leur temps et particulièrement à mes deux directeurs de thèse, Jean-Louis Giavitto et Dominique Pradelle.

... À Arshia Cont, Jean-Louis Giavitto, José Echeveste et Philippe Cuvillier pour leur magnifique travail sur Antescofo. Pour leur patience, leur ouverture et pour le beau langage qu'ils nous offrent à nous, compositeurs. En espérant pour celui-ci la plus longue des vies.

... À l'équipe Représentations Musicales de l'Ircam ainsi que l'équipe-projet Mutant pour leur formidable accueil.

... À Philippe Manoury, Yan Maresz et Marco Stroppa.

... À Françoise et Jean-Philippe Billarant, pour leur soutien, pour leur confiance et pour *Namenlosen*.

... À Jérôme et à Carmine, Mikhail, Markus, Thomas, Pascale, Deborah, Serge, Luca, Thomas, José-Miguel, Lara, Jean, Eric, Thibaut, Sylvie... et toutes les personnes que j'ai eu la chance de côtoyer dans les départements de recherche et de production de l'Ircam.

... À Alain Poirier et Gérard Assayag à qui je dois ma venue à l'Ircam. À François Roux, qui m'avait conduite jusque là.

... À G.D.H.

... À Jean-Louis Giavitto, pour son soutien indéfectible, son insatiable curiosité, sa gentillesse en toute circonstance, pour les milliers d'heures de discussions, d'envolées lyriques, de débats et de partage. Pour m'avoir supportée, dans tous les sens du terme. Pour les nims, les handlers, les fenêtres temporelles, les siouxeries antescofiennes en tout genre, la topologie, les cours de diplomatie, les gâteaux, la morphogène, la camomille, les images et innombrables métaphores, les mille et un conseils avisés. Pour avoir *soulevé* tant de choses.

... À mes parents et à ma sœur.



# TABLE DES MATIÈRES

INTRODUCTION	I
ORGANISER LE MULTIPLE	7
DE LA MULTITUDE	II
Stratégies compositionnelles	II
Modèles combinatoires et approches globales	II
Modèles processuels et modèles génératifs	12
Algorithmique, modèles Markoviens et fractales	12
Spectralisme	13
Organismes d'Information Musicale	14
Entre deux pôles, un spectre	14
Aide à la composition	14
Du paramétrique au relationnel	16
La notion de Denkraum	16
Classification et montage : du Denkraum à l'Atlas	18
ESPACES COMPOSITIONNELS	20
Topologie algébrique	21
Musique et topologie	21
Complexe simplicial	22
Q-analyse : la représentation de relations binaires	23
Relations binaires	23
Représentation des relations binaires par un complexe simplicial	24
Représentation générale de relations n-aires	25
Représentation spatiale et organisation du matériau compositionnel	25
Méthode de construction d'un espace	26
Qu'est-ce qu'une cellule ?	27
Composantes paramétriques	27
Du 0-simplexe au complexe simplicial	29
Quatre sous-espaces	31
Du local au global	32
Parle-t-on de distance ?	33
La nécessité du symbole	33
Parcours et directivité : la relation temps/espace	35
Des espaces autres	36

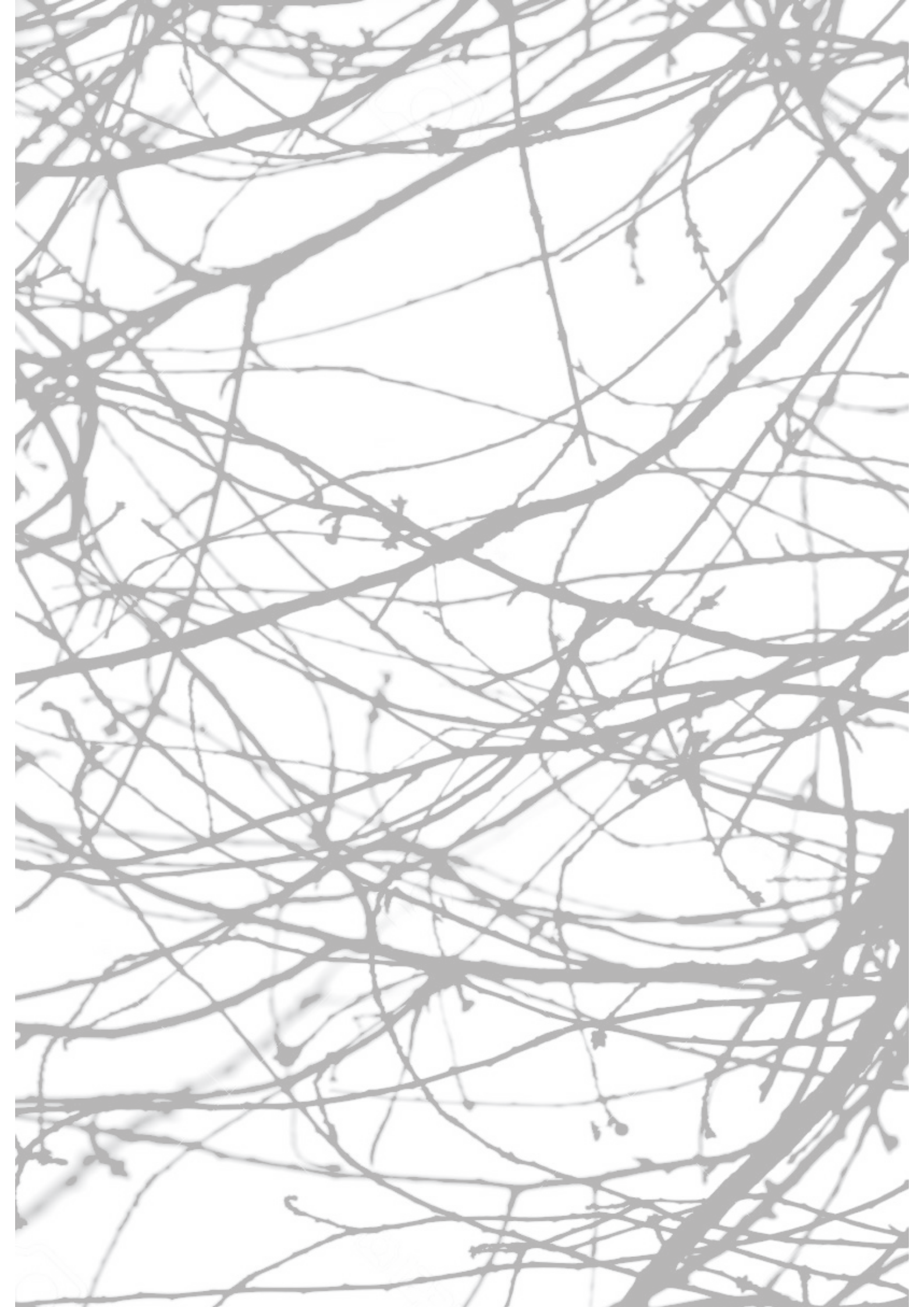


De l'imagination des formes.....	40
Mort et transfiguration .....	41
Applications pratiques : Sortir du noir et Namenlosen.....	42
Un exemple d'interprétation de cellules dans Sortir du noir .....	47
Quelques prolongements extra-musicaux... ..	50
<b>LA DIALECTIQUE DES TEMPS</b> .....	53
Multiplicités des temps en composition.....	54
Hors-temps, aide à la composition et mise-en-temps.....	54
Travail du son, timeline et chronomètre.....	55
Particularités de la musique de synthèse .....	56
La notion de qualité de temps.....	57
Concordance des temps.....	57
Organisation interne à une voix.....	58
Organisation des voix et de leurs synchronicités .....	59
Organisation macroscopique des temps.....	60
De la structure interne au composé temporel .....	61
Interdépendance des concepts de durée et de simultanéité .....	62
Relations, intrications, entrelacs, devenirs .....	62
<b>ORGANISER LE TEMPS</b> .....	65
<b>ANTESCOFO ET LES TEMPS MULTIPLES</b> .....	69
Une introduction au langage Antescofo .....	69
Actions et évènements .....	70
Actions composées .....	71
Processus .....	72
Les motifs temporels .....	73
Tempo et pulsation.....	73
Variables utilisateurs .....	73
Abstractions .....	74
Du déclenchement à la synchronisation .....	75
Feuilletages et réseaux temporels.....	77
Types temporels .....	77
Polyphonie, liaisons et interactions entre couches temporelles .....	79
Entrelacs électroniques .....	82
Les sentiers qui bifurquent .....	84
Les fenêtres temporelles .....	86
Le temps « local » du geste.....	86
L'expressivité au service du temps .....	87
Une fenêtre de temps... ..	88
Stratégies de synchronisation : le temps du phrasé.....	89
Stratégies de synchronisation avec tempo non-adaptatif.....	90

Stratégie lisse (loose) .....	90
Stratégie événementielle (tight) .....	91
Stratégie anticipante (ante).....	91
Dans la pratique.....	92
Stratégies anticipatives d'adaptation du tempo.....	92
Cibles statiques.....	92
Cibles dynamiques .....	94
Vers un phrasé électronique ?.....	96
« Mise en temps » d'un système par contraintes .....	97
Matrices de successions .....	97
Générateur de séquences.....	98
Mise en temps .....	99
Une taxinomie des échelles de temps.....	100
<b>NOTATION ET EXPRESSIVITÉ</b> .....	<b>101</b>
La nécessité du symbole.....	102
De la création « d'instruments » à la spécification de leurs « modes de jeu » .....	102
Une bibliothèque pour la spatialisation.....	105
« Je déballe ma bibliothèque » .....	108
Quelques réflexions sur différentes notations.....	109
Représentations graphiques dans Antescofo .....	110
<b>AU-DELÀ DE LA RÉTROACTION</b> .....	<b>114</b>
Organiser le multiple pour organiser le temps ? .....	115
<b>CONCLUSION</b> .....	<b>117</b>
<b>ANNEXE I</b> .....	<b>123</b>
<b>IMPLÉMENTATION MATHEMATICA DES ESPACES DE NAMENLOSEN</b> .....	<b>124</b>
Complexes simpliciaux — construction.....	125
Namenlosen .....	126
Bibliothèque d'enveloppes.....	126
Dictionnaire timbre .....	127
Dictionnaire timbre .....	128
Déclaration des espaces.....	132
Fonctions .....	133
Affichage d'un espace .....	133
Filtrage d'un espace .....	133
Espace Mund1 .....	134
Espace Mund2.....	135
Espace Mund3.....	136
Espace Mund4.....	137

SORTIR DU NOIR	138
THÉÂTRE D'OMBRES	139
ANNEXE 2	141
ANTESCOFO : USER GUIDE	143
Interactive Music Systems.....	143
The Antescofo approach: coupling score following with a programming language.....	143
Brief history of Antescofo .....	144
Structure of an Antescofo Score .....	144
An interweaving of musical events and electronic actions .....	145
The file structure of an Antescofo augmented score .....	147
Elements of an Antescofo Score .....	147
Simple identifiers: Antescofo keywords and reference to the host environment .....	148
@-identifiers: Functions, Macros, and Attributes .....	149
\$-identifiers : Variables and Parameters .....	149
Events .....	150
Event Specification .....	150
Events as Containers .....	152
TRILL .....	152
MULTI.....	152
Compound Events .....	153
Event Attributes.....	153
Event Label.....	154
The @modulate Attribute .....	154
Actions in Brief.....	154
Delays.....	154
Label .....	156
Action Execution .....	156
A Brief overview of Antescofo features .....	156
A useful action : the curve .....	157
Make your life easier with macros ! .....	157
Tour the loop .....	158
Build your own world.....	159
Why do you need data structures... ..	159
In processes we trust .....	160
A conditional world.....	161
Become the time master.....	162
Antescofo Workflow.....	163
Editing the Score .....	163
Importing Scores to Antescofo (import of Midi files and of MusicXML files).....	164
Importing MIDI scores to Antescofo .....	164
Importing MusicXML scores to Antescofo .....	164

Using AscoGraph .....	165
The App Menu .....	166
Color Scheme .....	166
Interaction Between Visual and Text Editors in AscoGraph .....	166
Shortcuts .....	167
Edit your curves .....	167
Automatic Filewatch: Using Another Text Editor .....	168
Styling your score .....	169
<b>Beyond score following.....</b>	<b>169</b>
Antescofo as a sequencer.....	169
Hierarchical scores .....	170
Open scores and installations.....	170
Beyond Max and Pure Data.....	171
A SuperCollider example.....	171
A CSound example.....	172
OSC.....	172
Be adventurous !.....	172
<b>ANNEXE 3</b> .....	<b>175</b>
Bibliothèque de spatialisation.....	177
A tutorial for composers .....	182
<b>BIBLIOGRAPHIE</b> .....	<b>187</b>
<b>PORTFOLIO</b> .....	<b>191</b>



# INTRODUCTION

*« Pour savoir il faut donc se tenir dans deux espaces et dans deux temporalités à la fois. Il faut s'impliquer, accepter d'entrer, affronter, aller au coeur, ne pas louvoyer, trancher. Il faut aussi — parce que trancher l'implique — s'écarter, violemment dans le conflit, ou bien légèrement, comme le peintre lorsqu'il écarte de sa toile pour savoir où il en est de son travail. On ne sait rien dans l'immersion pure, dans l'ensoi, dans le terreau du trop-près. On ne saura rien, non plus, dans l'abstraction pure, dans la transcendance hautaine, dans le ciel du trop-loin. Pour savoir il faut prendre position, ce qui suppose de se mouvoir et de constamment assumer la responsabilité d'un tel mouvement. Ce mouvement est approche autant qu'écart : approche avec réserve, écart avec désir. Il suppose un contact, mais il le suppose interrompu, si ce n'est brisé, perdu, impossible jusqu'au bout. »<sup>1</sup>*

La recherche que nous présentons ici s'est articulée autour de deux axes distincts qui, au fil du temps et des recherches, se sont trouvés imbriqués. Le premier concernait l'organisation du matériau et notamment le processus compositionnel conduisant à sa singularisation. Avant même d'avoir débuté cette thèse, nous avons imaginé et réalisé une représentation du matériau par un système de graphes, symbolisant les liens existant entre les divers éléments utilisés pour la composition d'une pièce. Il s'agissait d'un modèle rudimentaire qui ne faisait sens que pour nous. Notre arrivée à l'Ircam et la rencontre avec Jean-Louis Giavitto — qui deviendra ensuite le directeur de cette thèse — a été décisive, notamment par la découverte d'un champ de recherche sur lequel celui-ci avait travaillé avant sa venue à l'Institut : la topologie et la programmation spatiale. Cette recherche a donc été pour nous l'occasion de créer un nouveau modèle beaucoup plus pertinent et a révélé certaines problématiques qui se sont avérées majeures pour notre travail de composition. Il s'agissait de traiter de la nécessité d'organiser des espaces compositionnels ne se limitant pas à la production plus ou moins volumineuse d'un matériau mais pouvant intégrer, dans leurs structures internes, des éléments d'échelles supérieures permettant, sinon de faire disparaître, au moins de rendre plus poreuse la frontière érigée entre les notions de forme et de matériau.

Le second s'appliquait à l'écriture de l'électronique dans le cas particulier de la musique mixte. Les années passées à travailler dans l'équipe Mutant<sup>2</sup> qui développait le langage Antescofo — dédié à l'origine à la musique mixte — nous ont permis de développer nos recherches, progressivement focalisées sur la question des temps multiples. Il importait pour nous d'atteindre des niveaux d'expressivité (notamment puisqu'il s'agissait d'un langage textuel) qui permettent d'amener l'écriture de l'électronique à un niveau équivalent à celui de l'écriture instrumentale. Si nous ne considérions pas l'électronique comme un simple instrument à ajouter au parc instrumental à la disposition des compositeurs, nous voulions en revanche pouvoir avoir les mêmes exigences en terme de précision et d'organisation du discours. Par ailleurs, les spécificités du langage Antescofo pouvaient nous fournir des outils aptes à faciliter la définition de processus dont les particularités temporelles auraient été extrêmement délicates à aborder sans leur concours.

1 Georges Didi-Huberman. *Quand les images prennent position*. T. *L'oeil de l'histoire*, 1. Les Éditions de Minuit, 2009.

2 L'équipe Mutant faisait partie intégrante de l'équipe Représentations Musicales à l'Ircam et était une équipe-projet Inria. Le travail de l'équipe s'est focalisé sur le langage Antescofo, sur sa machine d'écoute et son langage réactif.

*... Avertissements ...*

Cette thèse de composition contient des aspects techniques, inhérents notamment à l'utilisation d'un langage synchrone et d'un modèle lié à la topologie algébrique. Ces aspects techniques font pour nous partie intégrante de notre travail compositionnel, ils donnent — tout du moins selon nous — corps à la dimension « art-science » de cette thèse, non pas vue comme étendard ou comme justification, mais comme moyen d'amener notre recherche artistique au-delà des modèles classiques. Chaque outil utilisé, chaque élément d'ordre technologique ou extra-musical, n'est présenté ici que parce qu'il rend *lisible* quelque chose qui, d'après nous, n'aurait pu l'être autrement. De ce fait, ce travail de recherche ne peut-être autre chose qu'une prise de position artistique qui nous amènera à parler, de façon plus ou moins explicite, d'esthétique, de temps musical, de rapports dialectiques entre forme et matériau ou encore d'écriture électronique.

Avant d'aller plus loin il nous incombe de préciser quelques éléments quant à notre rapport à ce qui est communément appelé « la technologie ». Cette thèse a été réalisée à l'Ircam, institut de recherche alliant historiquement scientifiques et compositeurs. Il nous semble vital de rappeler que si la technologie fait partie intégrante de notre espace de travail et des diverses créations qui ont été réalisées durant cette thèse, nous n'avons en revanche aucune déférence à avoir vis-à-vis de celle-ci. La technologie n'a d'intérêt pour nous que dans la mesure où elle est pourvoyeuse d'un surplus de savoir, de points de vue qui puissent nous faire envisager un objet *autrement* et d'outils qui nous permettent d'envisager la composition avec un regard nouveau nous amenant éventuellement à reconsidérer certains de ses aspects. Il nous échoit également de prendre garde à ne pas nous contenter des utilisations anecdotiques, de celles qui, si elles portent aux nues une éphémère nouveauté, n'amènent pas avec elles ce qu'il faut de remise en cause, de nouvelles problématiques et, plus important encore, demeurent exogènes au discours musical.

Si les différentes pièces composées durant cette thèse seront évoquées lorsqu'il sera nécessaire d'avancer un exemple pratique, nous n'avons pas souhaité consacrer de chapitre traitant exclusivement d'une pièce en particulier, par un processus d'auto-analyse. Il nous semblait plus fécond d'irriguer des propos théoriques, esthétiques ou techniques par des illustrations qui, sur le moment, pouvaient venir éclairer le propos ou tout du moins le rendre moins aride. Les deux axes de recherche dont nous avons parlé ont été les piliers de notre travail de création. Si le mariage du temps de la composition/production et de la recherche n'est pas toujours évident, il n'en demeure pas moins que les deux se sont nourris et ont pu mutuellement se remettre en question.

Enfin, cette thèse est, par la force des choses, d'un format pour le moins non standard. Elle n'est en rien un travail musicologique et, malgré le fait qu'elle soit intégrée à une école doctorale d'informatique, n'est pas davantage une thèse scientifique. Qu'est-ce alors que cet objet ? Sans nul doute une recherche chevillée au corps du travail compositionnel qui, faisant parfois appel à des disciplines extra-musicales lorsque cela lui était nécessaire, s'est trouvée métamorphosée en créature tentaculaire. Il nous importait d'opter, non pas pour une prise de parti, mais pour une prise de position artistique qui mette tout en œuvre pour tâcher de répondre aux questionnements qui l'habitaient. Si les réponses ne sont pas toujours venues, la recherche n'en a pas été moins fertile. Il nous paraissait nécessaire de tenter — d'essayer tout au moins — de théoriser un certain nombre de concepts qui nous semblaient pertinents, de mettre à l'épreuve de la création certaines idées parfois jusque là restées à l'état d'ébauches de pensée.

*Linéament*

La technologie offre aux compositeurs des outils et des langages permettant d'envisager de nouvelles approches et pratiques musicales, tant du point de vue de la conception du matériau que de celui de son articulation à la forme. Dans la musique mixte, l'avènement du « temps réel » — de la performance prise comme modèle à construire — entraîne avec lui de nouvelles possibilités d'écriture du temps et un rapport particulier aux processus qui le constituent.

Si les questions du temps, du matériau, de la forme et de leurs relations sont des questions centrales — et depuis bien longtemps — pour les compositeurs, il importe de se demander si la vieille dichotomie entre forme et matériau n'est pas à repenser en regard des nouveaux dispositifs susceptibles d'amener une appréhension à la fois plus fine et plus globale des diverses dimensions temporelles. Dans un tel contexte, et dans le cas plus précis de la musique mixte, une question a parcouru — pour ne pas dire hanté — toute notre recherche. Celle-ci consistait à se demander dans quelle mesure l'existence d'un continuum entre les notions de forme et de matériau était liée à la composition consciente d'entrelacs temporels ? Cette question impliquait l'utilisation de paradigmes capables de nous faire appréhender les nombreuses composantes qui font l'identité d'un matériau ou d'une forme, et de nous faire percevoir les liens qu'elles entretiennent mutuellement. Elle nécessitait également un langage permettant de parler de façon expressive des différents temps avec lesquels nous composions.

Il existe aujourd'hui beaucoup d'outils permettant de générer un matériau, d'analyser un son et d'en donner des descriptions variées, parfois réutilisées par la suite pour produire un nouveau matériau. Il nous est apparu que l'immense majorité de ces outils et des paradigmes utilisés actuellement gravitaient autour d'une pensée essentiellement paramétrique. Pensée qui pouvait engendrer des résultats radicalement différents suivant le compositeur qui la mettait en œuvre mais qui, selon nous, avait la fâcheuse tendance à séparer par cloisonnement respectif « l'alpha et l'oméga » du travail compositionnel, à savoir la forme et le matériau. Pour cette raison, il nous a semblé pertinent d'opérer une sorte de glissement de la pensée, qui nous permette de pouvoir saisir un matériau non pas par un ensemble de paramètres pris individuellement mais par un ensemble de relations, de liens existants entre les différentes entités musicales avec lesquelles nous travaillions. Il nous paraissait que les notions de qualité de temps ou de forme ne pouvaient s'appréhender dans leur complexité qu'avec le concours de plusieurs échelles de temps comportant un certain nombre de réseaux de dépendances. Cela impliquait donc naturellement de nous tourner vers *une pensée relationnelle*.

Cette réflexion est, d'une certaine manière, à l'origine des deux axes qui ont formé notre recherche compositionnelle. En premier lieu, nous nous sommes intéressés à un type particulier de représentation spatiale, les structures topologiques, qui étaient ici utilisées pour figurer l'organisation d'un matériau défini par le compositeur. Nous partions d'un postulat relativement simple : si deux choses interagissent, alors on peut considérer qu'elles sont voisines dans un certain espace. Nous verrons tout au long de cette thèse que certaines notions reviendront comme des leitmotifs, en contrepoint de cette pensée relationnelle : l'entrelacs, l'intrication des différentes échelles de temps qui composent une œuvre musicale mais également la notion de *singularisation*.

En second lieu, notre pratique de la musique mixte nous a conduit vers l'apprentissage du langage Antescofo, langage synchrone dont l'expressivité et les possibilités d'écritures temporelles se sont résolument perfectionnées durant ces cinq dernières années. En plus d'offrir une panoplie de possibilités en terme d'écriture des différents temps, le langage nous permettait d'explicitier certaines relations et de composer des processus complexes qu'il aurait été laborieux de reproduire dans les logiciels habituellement utilisés en musique mixte. De surcroît, Antescofo nous a permis d'amener l'écriture de l'électronique bien au-delà de ce que nous avons pu faire auparavant, intégrant notamment des notions de phrasé liées aux stratégies de synchronisations récemment implémentées dans le langage, ou encore des fonctions permet-



tant de ne pas être en permanence lié au suivi de partition grâce à l'intégration de variables externes.

Si il existe un continuum entre forme et matériau, alors nous devons nous emparer de tous les moyens en notre possession qui nous permettront d'imaginer, de réfléchir et de composer les entrelacements temporels qui le constitue. Qu'il s'agisse pour cela d'un langage permettant de spécifier toutes les subtilités temporelles d'un ensemble de lignes et de processus plus ou moins liés entre eux ou qu'il s'agisse d'environnements compositionnels nous permettant d'appréhender la diversité des liens constitutifs d'un certain matériau.

L'espace de pensée musicale dont nous allons parler tout au long de cette thèse à travers la notion d'espace compositionnel est pour nous un lieu de tressage des différentes couches temporelles qui l'habitent. Par cela il est pour nous un lieu d'émergence de formes dans lequel la pensée relationnelle tient une place fondamentale.

### *Organisation du manuscrit*

Le présent manuscrit se divise en deux grandes parties, figurant les deux axes évoqués plus avant. La première concerne l'organisation du multiple, la manière dont la multitude d'éléments composant le matériau musical peut s'organiser de façon à se lier à des questions d'ordre formel. Après un tour d'horizon très rapide des différentes stratégies compositionnelles mises en place par quelques compositeurs emblématiques, nous présentons un modèle de pensée introduit en histoire de l'art et qui a influencé certains de nos choix de recherche, en particulier en ce qui concerne la pensée relationnelle. Nous présentons un modèle issu de la topologie algébrique destiné à créer des espaces compositionnels qui puissent être des espaces de pensée, d'édification d'un discours et de qualités de temps qui émergent au sein d'un réseau de dépendances. Ces espaces ont notamment été utilisés dans les diverses pièces composées durant cette thèse. Celles-ci seront évoquées tout au long du manuscrit, de manière à éclairer le propos et à donner des applications pratiques. Ce modèle nous amènera à reconsidérer certaines notions : le rapport local/global, les notions de voisinage, de parcours et de distance, ou encore la relation temps/espace dans une optique de directivité.

Nous clôturons cette partie par un chapitre dédié à la dialectique des temps, d'abord dans la pratique puis au sein même de la composition. Il s'agira notamment de questionner la manière dont une qualité de temps peut être définie. La construction d'un temps particulier relève de la constitution d'un complexe de temporalités synchronisées entre elles de manière à donner une identité temporelle particulière à cette construction. Les réflexions liées aux espaces compositionnels, particulièrement les relations entre échelles de temps, trouveront dans ce chapitre un écho certain et viendront former une sorte de fil rouge qui parcourra l'ensemble de ce manuscrit.

La seconde partie aborde la question de l'organisation temporelle dans le contexte particulier de la musique mixte et du langage Antescofo. Une introduction au langage est donc présentée et peut être complétée par la lecture d'un *User guide* rédigé pendant la recherche, exposé ici en annexe. Nous développons ensuite, en continuité avec les propos tenus en première partie, une réflexion menée à travers un certain nombre d'exemples concrets sur les polyphonies et réseaux temporels. En nous focalisant sur une échelle de temps relativement réduite, nous présentons un travail fait autour des fenêtres temporelles, qu'elles soient utilisées pour dessiner les contours du phrasé d'un passage électronique ou qu'elles constituent une entité temporelle indépendante qui puissent malgré tout être synchronisées au jeu d'un instrumentiste. Nous exposons ensuite les différents types de stratégies de synchronisation qui, si ils sont choisis de manière à correspondre au contexte musical dans lequel ils s'inscrivent, permettent d'appliquer à l'électronique un certain degré de fluence amenant à spécifier un type de phrasé appliqué de façon plus ou moins locale. Un exemple de système par contraintes implémenté en Antescofo nous permettra de mettre en avant un aspect du langage relativement peu utilisé et de faire un lien direct avec nos espaces compo-

## INTRODUCTION

tionnels par l'implémentation d'une matrice d'intervalles et d'un générateur de séquences directement lié à la constitution des espaces présentés en première partie. Pour revenir à nos préoccupations temporelles, nous établissons une taxinomie des échelles de temps dans Antescofo.

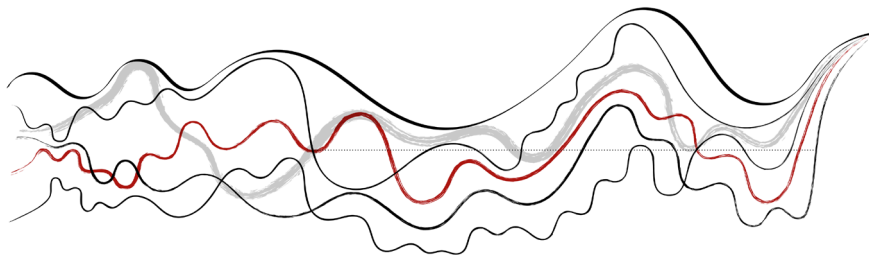
La question de la notation et de l'expressivité est ici traitée dans le contexte précis du langage utilisé. La notation dédiée à la musique électronique étant un sujet à part entière que nous ne pouvons traiter ici, nous nous focalisons sur des réflexions d'ordre plus général et sur l'expressivité du langage Antescofo. À cette occasion nous présentons une bibliothèque dédiée à la spatialisation et implémentée pour la réalisation de la pièce *Namenlosen* ainsi qu'une méthode d'implémentation destinée à définir des ensembles de modes de jeu attribués à des générateurs de synthèse. Nous concluons cette deuxième partie par quelques réflexions sur la musique mixte et sur le dépassement du modèle classique dans lequel les relations unilatérales restent souvent dominantes.

*Ami lecteur, en entrant, n'oublie pas ta lampe de poche.  
Le brouillard ne s'est pas encore tout à fait dissipé.*



# PARTIE I

## ORGANISER LE MULTIPLE



*« L' "en soi" est même une conception absurde : une "modalité en soi" est un non-sens : le concept de l' "être", de la "chose", n'est pour nous toujours qu'un concept de relation...*

*[...]*

*... Le monde, abstraction faite de notre condition d'y vivre, le monde que nous n'avons pas réduit à notre être, notre logique et nos préjugés psychologiques n'existe pas comme monde en "soi"; il est essentiellement un monde de relations, regardé à un point différent, il prend chaque fois visage nouveau : son être est essentiellement différent à chaque point ; il appuie sur chaque point, chaque point lui résiste - et ces additions sont dans chaque cas parfaitement incongruentes. »<sup>1</sup>*

---

1 F. Nietzsche, *La volonté de puissance*. 293.

Si le XX<sup>ème</sup> siècle a fait exploser tous les cadres, il a notamment mis au devant de la scène la profusion des paramètres musicaux mis à la disposition des compositeurs, de par la disparition plus ou moins progressive des restes du langage tonal et des hiérarchies qu'il sous-tendait mais également par un certain nombre de découvertes et de recherches, notamment en informatique, en acoustique et psycho-acoustique permettant de révéler « la mécanique » — toute non-linéaire puisse-t-elle être — du son et d'inventer des langages permettant aux machines d'en générer. À l'heure où les moyens de générer du matériau se sont significativement multipliés et où la question du matériau en lui-même a largement été traitée, nous avons souhaité aborder celle de son organisation interne et de ses liaisons à la forme. L'idée était, en arrière-fond, de situer le « lieu », tout du moins dans notre propre expérience, où le matériau se singularise et où les conditions de possibilité d'une forme sont susceptibles d'émerger.

Si les diverses utilisations paramétriques du matériau sont aujourd'hui relativement avancées, elles demeurent souvent circonscrites à l'objet qu'elles traitent, quitte à construire d'immenses étagères à paramètres sur lesquelles on empile des données et processus plus ou moins élaborés. Or, il nous semble ici fondamental d'opérer un changement de paradigme en passant d'une pensée majoritairement paramétrique à une pensée de la relation, qui puisse notamment « tenir » face à la multitude des informations à traiter.

Nous le verrons tout au long de cette partie, une pensée relationnelle implique un certain nombre de présupposés quant à la façon d'envisager l'organisation du matériau. Cela impliquera des changements d'échelles fréquents et des modes de représentation parfois étrangers au monde musical. Si les solutions trouvées ici, parfois toutes provisoires, nous sont tout à fait personnelles, il nous apparaît cependant qu'elles mettent en lumière et questionnent cette pensée musicale relationnelle que nous appelons de nos vœux.

Nous terminerons cette première partie par une réflexion sur la multiplicité des temps — que ce soit dans le travail mais aussi dans la conception même d'une œuvre musicale — en tentant de la mettre en vis-à-vis d'une réflexion sur les espaces compositionnels que nous aurons tenu plus avant.

## DE LA MULTITUDE

Stratégies compositionnelles .....	11
Aide à la composition .....	14
La notion de <i>Denkraum</i> .....	16

## ESPACES COMPOSITIONNELS

Topologie algébrique .....	21
La nécessité du symbole .....	33
Parcours et directivité : la relation temps/espace .....	35
Des <i>espaces autres</i> .....	36
Applications pratiques : <i>Sortir du noir</i> et <i>Namenlosen</i> .....	42
Quelques prolongements extra-musicaux.....	50

## LA DIALECTIQUE DES TEMPS

Multiplicités des temps en composition .....	54
La notion de qualité de temps .....	57

## ORGANISER LE MULTIPLE : DE LA MULTITUDE

# DE LA MULTITUDE

« *Il est des nébuleuses qu'aucun œil ne distingue.* »<sup>1</sup>

L'analyse du son, le développement de théories musicales abandonnant les restes d'héritage tonal, la musique électronique, l'apport de modèles d'origine scientifique ou encore l'intégration de modes de jeu étendus pour les instruments acoustiques ont été des éléments clés de l'évolution des langages jusqu'à nos jours et notamment de la multiplication des paramètres à prendre en compte. Cette multiplication extraordinaire des paramètres et leur intégration au cœur du travail compositionnel, a été à la fois rendu nécessaire et rendu possible par le développement des technologies. Cependant, il nous semble qu'il s'est souvent focalisé sur des échelles réduites, éludant les questions d'ordre formel qui restent pourtant essentielles et tout à fait centrales.

Si l'époque récente, celle du début du XXI<sup>ème</sup> siècle, semble moins portée sur la création de systèmes ou de langages, il nous semble que les questions restent, malgré tout, sensiblement les mêmes, nécessitant de ce fait, une remise continuelle sur l'établi. Face au problème crucial de la forme, va se mettre en place un spectre assez large de stratégies allant d'un contrôle radical du détail (au risque de perdre la grande forme) à la réalisation d'une vision d'ensemble (au prix d'un abandon potentiel d'une détermination rigoureuse du détail). Sur ce constat, nous discuterons dans ce chapitre de la nécessité du passage d'une pensée paramétrique, assez prédominante, à une pensée relationnelle.

## Stratégies compositionnelles

Nous n'exposerons pas ici de liste exhaustive des pratiques compositionnelles. Nous nous contenterons d'en sélectionner quelques unes qui nous semblent importantes à présenter dans une perspective de génération et d'organisation du matériau et des éventuelles conséquences formelles qu'elle peut éventuellement avoir. Ceci pour mettre en perspective notre propre travail et mettre en exergue les nécessités qui nous ont amené à développer notre modèle propre.

### Modèles combinatoires et approches globales

On connaît la prééminence accordée aux paramètres pris indépendamment dans la musique sérielle. Celle-ci s'est appliquée à diverses échelles et de manières différentes selon les compositeurs. La cohérence repose par exemple chez **Boulez** sur « des déductions localisées et variables »<sup>2</sup>. Hauteurs, durées, intensités et timbres (dimensions auxquelles le compositeur ajoutera une cinquième : l'espace) seront combinés, parfois associés et élargis à des champs et au bruit qui s'intègre alors à la construction formelle. L'œuvre est organisée selon des critères de définition et de sélection — opérant selon un degré de fixité ou de changement — et des critères de combinaison et d'arrangement — soit l'organisation syntaxique du langage. « La forme n'est plus pensée et vécue comme un geste, mais comme un *ensemble conceptuel* »<sup>3</sup>, une mise en structure de structures locales engendrées par un principe sériel et des méthodes de prolifération. La conclusion de *Penser la musique aujourd'hui* pose la question cruciale du sens<sup>4</sup>. L'articulation de l'ordre

1 W. G. Sebald, *Les émigrants*. Actes Sud, Babel, 2001.

2 Pierre Boulez, *Penser la musique aujourd'hui*. Gallimard, collection Tel n°124, 1987. p43.

3 Laurent Feneyrou, *Pierre Boulez*. Dans *Théories de la composition musicale au XX<sup>ème</sup> siècle*. Symétrie Recherche, 2013.

4 « à toutes ces méthodes, il faut donner un sens », p166.



syntactique de l'écriture et de l'ordre sémantique de la perception est en jeu et sera présenté par la suite dans ses cours au Collège de France comme dénotée par la notion de *thème*<sup>5</sup>.

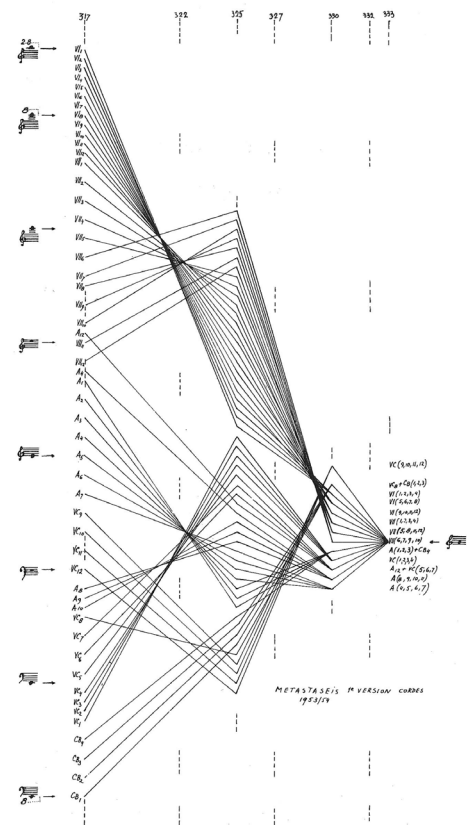
Chez **Stockhausen**, il y a un idéal d'unité et d'organisation totale, à tous les niveaux. Le compositeur parle à ce propos de « faire régner une loi universelle sur tous les domaines temporels »<sup>6</sup>. La corrélation sérielle entre hauteurs et durées s'appuie non plus sur des unités hypothétiquement perceptibles, mais sur des proportions entre quantités globales (« champs de temps »). « Le principe de *continuité* s'impose chez Stockhausen comme une donnée fondamentale expliquant l'évolution de la musique ponctuelle à la composition sur formule »<sup>7</sup>. Il y a d'une part une pensée de la division de phase et de l'autre une graduation chromatique quasi systématique visant à remplacer le temps comme discontinuité par un continuum relevant d'un *ordre supérieur*, à savoir le temps sériel. Avec le concept de formule, il y a de surcroît l'idée d'engendrement formel par *croissance organique* et d'ordonnement de *qualités sonores* (on pense ici en premier lieu à la pièce *Mantra*). La formule est donc un organisme (et non un thème) à l'image duquel s'articule le tout. Si les critères morphologiques peuvent s'apparenter aux dichotomies boulezienne (« morphe-amorphe » et « statique-analytique » chez Boulez, « processus-état » et « configuration-structure » chez Stockhausen), ils fonctionnent sur une distribution par complémentarité-parenté qui postule la subordination des parties à une unité supérieure, ce qui est tout à fait étranger aux conceptions bouleziennes. L'idée de trajectoire dramatique est ici totalement récusée.

## Modèles processuels et modèles génératifs

### Algorithmique, modèles Markoviens et fractales

**Xenakis** défend une approche globale du phénomène sonore (notamment par une fusion des dimensions traditionnelles : hauteur, rythme, intensité) où la forme est un tout que l'on compose en emboîtant différentes sections. La texture musicale est de ce fait pensée comme une intégration totale des sons qui la composent (d'où l'importance de la notion de masse). Les graphiques, très utilisés, respectent les coordonnées classiques de la notation solfégique que sont le temps et la hauteur. Il y a ici association de données géométriques (une position en *x* et en *y*) à des paramètres musicaux (voir l'image ci-contre). S'intéressant d'avantage aux masses qu'à l'individualité des mouvements de sons<sup>8</sup>, Xenakis va développer, via le calcul probabiliste, ce qu'il nommera la « musique stochastique ». La théorie des cribles lui servira également pour générer des échelles. Les modèles mathématiques sont ici utilisés de manière à générer un matériau mais également à gérer l'évolution du discours par associations directes. Il y a l'idée d'une certaine mécanique, d'une « boîte noire » lorsque la machine est employée, qui produirait une œuvre entière après introduction de données initiales. Représentant d'une pensée musicale formalisatrice, il fût l'un des premiers compositeurs à utiliser l'ordinateur comme outil d'aide à la composition.

*Metastaseis* (1953-54), mesures 317-333 : graphique de Xenakis  
Source : Iannis Xenakis, *Musique. Architecture*, Tournai, Casterman, 1976, p. 8



5 Pierre Boulez, *Leçons de musique. Points de repère III*. Éditions Christian Bourgeois, 2005. p203-336.

6 Karlheinz Stockhausen, *Texte zur elektronischen und instrumentalen Musik. Texte zur Musik*, vol. I., édition Dieter Schnebel. 1963. p. 211-221.

7 François Decarsin, *Karlheinz Stockhausen, dans Théories de la composition musicale au XX<sup>ème</sup> siècle*, vol.2. Collection Symétrie Recherche, série 20-21, 2013. p.1021-1039.

8 Iannis Xenakis, *Musique Architecture*. Tournai : Casterman. 1971.

**Philippe Manoury** fait appel depuis longtemps à des processus lui permettant d'engendrer des structures comportant une part d'aléatoire : les chaînes de Markov. Utilisées notamment pour contrôler des successions d'intervalles, ces chaînes ont pour but d'engendrer des séquences aléatoires qui soient néanmoins identifiables. « Il faut admettre que l'ordre de succession des événements musicaux est d'une importance aussi grande que les événements eux-mêmes. C'est lui qui donne un "sens" à ce que nous écoutons. [...] Ce que produisent les matrices de Markov n'est ni plus ni moins qu'une partition virtuelle qui se crée en temps réel et se régénère d'elle même »<sup>9</sup>. Les matrices de Markov sont là pour générer un matériau infini, un flot ininterrompu que l'on « guide » vers telle ou telle direction grâce à des probabilités.

Les grammaires musicales génératives, elles, sont principalement utilisées dans le but de trouver une cohérence dans l'ordre temporel d'enchaînement d'événements musicaux. À partir d'éléments morphologiques qui vont servir de critères « thématiques » (les *invariants formels*<sup>10</sup>) il s'agit de faire émerger une cohérence dans la continuité du discours musical.

Chez **Alberto Posadas**, il ne s'agit pas d'établir une vision formaliste de la musique d'un point de vue mathématique (comme chez Xenakis par exemple) mais de modeler plusieurs de ses stratégies compositionnelles en s'appuyant sur des concepts et des outils propres à la science. Lorsqu'il utilise des fractales, principalement via des systèmes de Lindenmayer, celles-ci vont fédérer la construction temporelle — dans son quatuor *Liturgia Fractal* notamment — et lui permettre d'obtenir des réservoirs de hauteurs, battant par cela les piliers harmoniques et contrapuntiques. En essayant de contrôler une évolution progressive de son matériau par des courbes de Bézier avec des techniques semblables aux interpolations harmoniques de la musique spectrale, son intérêt pour les transformations graduelles le rapproche de la topologie différentielle.

### *Spectralisme*

Largement influencé par les travaux d'Abraham Moles, **Gérard Grisey** a repris le schéma de la théorie de l'information comme modèle des processus de communication de la musique (on notera notamment l'idée de la musique comme modulation du temps). « Grisey substitue aux catégories arbitraires et généralement dualistes, par lesquelles on est tenté de classer les durées (brève/longue, ternaire/binaire, valeurs rationnelles/irrationnelles, symétrie/asymétrie...), une échelle de complexité qui aurait le mérite de renvoyer directement aux phénomènes du temps musical tels qu'ils sont perçus, et de laisser voir une continuité niée par les répartitions de l'organisation hiérarchique des techniques néo-sérielles »<sup>11</sup>. Cette échelle va de l'ordre (périodicité et prévisibilité maximale) au désordre (statistique ou lisse, sans aucune prévisibilité), classant les phénomènes du simple au complexe selon un critère de prévisibilité des événements. Pas de raisons pour les compositeurs spectraux de distinguer le timbre de l'harmonie entre lesquels existe un continuum comparable à celui qu'on trouve entre rythme et fréquence. Il s'agit d'intégrer toutes les catégories du son, en insistant sur leurs *qualités*.

« Les différents processus de mutation d'un son en un autre son ou d'un ensemble de sons en un autre ensemble constituent la base même de mon écriture musicale, l'idée première, le gène de toute composition. Le matériau découle du devenir sonore, de la macrostructure, et non l'inverse. Autrement dit, il n'y a pas de matériau de base (cellule mélodique, complexe de sons, durées, etc.), dont la forme constituerait une sorte de développement a posteriori. C'est le processus qui est premier, c'est lui qui gère la mutation des figures sonores et qui amène à en créer sans cesse de nouvelles. »<sup>12</sup>

9 Philippe Manoury, *Les chaînes de Markov... à l'infini*. <http://www.philippemanoury.com/?p=5685>. 2015.

10 Philippe Manoury, *Les Grammaires Génératives musicales*. <http://www.philippemanoury.com/?p=5025>. 2012.

11 Angelo Orcalli, *La pensée spectrale*, dans *Théories de la composition musicale au XX<sup>ème</sup> siècle*, vol.2. Collection Symétrie Recherche, série 20-21, 2013. p.1511-1573.

12 Gérard Grisey, *Zur Entstehung des Klanges...*, *Darmstädter Beiträge zur neuen Musik*, 17 (1978). Traduction française

## *Organismes d'Information Musicale*

L'article publié par **Marco Stroppa** en 1988<sup>13</sup>, « Organismes d'information musicale : une approche de la composition » est intéressant à présenter ici, ne serait-ce que pour sa singularité.

*« Je rêvais d'une musique de morphologies s'interpellant dialectiquement les unes et les autres, et je cherchais des formes qui pouvaient les mettre en valeur. Au bout de cette expérience américaine, j'ai écrit un essai sur ma poétique musicale (Organismes d'information musicale : une approche de la composition) qui est encore important pour moi aujourd'hui. »*<sup>14</sup>

Un questionnement que pose le compositeur en introduction de son article, qui fera directement écho aux recherches que nous exposerons par la suite, est celui de se demander comment « décrire le fait que plusieurs éléments différents puissent être un seul exemple de la même identité et soient donc similaires ? ». Le terme organisme fait référence à une « entité dynamique et complexe dont l'évolution ne peut être expliquée ni prédite par des règles synthétiques » composée de plusieurs composants et propriétés qui vont entretenir certaines relations donnant lieu à une *forme* spécifique. L'organisme a une évolution *orientée* vers un but, incluant de ce fait une notion de trajectoire qui peut ainsi le distinguer d'un organisme apparemment similaire.

### **Entre deux pôles, un spectre**

Les diverses démarches compositionnelles de ces 70 dernières années ont marqué bien des antagonismes. Entre induction et déduction, entre local et global, entre « structuralistes » et « phénoménologiques », il semble que ce soit établi un spectre large, entre ces différents pôles, sur lequel les compositeurs ce sont positionnés, parfois temporairement pour se déplacer par la suite vers une position leur paraissant plus féconde. Les approches plus ou moins individuelles, illustrent le parti-pris d'individualités et correspondent à un style de travail.

Si les démarches sont différentes, elles semblent donc cependant toutes répondre, à leurs manières, à des questions inhérentes à celle du rapport entre matériau et forme et, par induction, à celle de la direction du rapport entre local et global. Il est d'ailleurs intéressant de constater qu'à partir d'une même technique, d'une même « école », les positions sur ces questions peuvent s'avérer radicalement différentes (l'exemple le plus criant est peut-être celui de Boulez et de Stockhausen).

Il faut noter, à ce stade de notre exposé, que les concepts de matériau et de forme sont souvent découplés ou sont liés de façon quasi mécanique. Il s'agit dans la majorité des cas de rapports de subordination : le matériau sur la forme, la forme sur le matériau.

## **Aide à la composition**

Les outils d'aide à la composition développés dans les années 90 ont permis à bon nombre de compositeurs de construire leur propre univers musical. Différents paradigmes de programmation ont été utilisés pour les langages de composition assistée par ordinateur (CAO). Mais le paradigme de programmation dominant dans les langages de CAO demeure celui de la programmation fonctionnelle. Les langages et

---

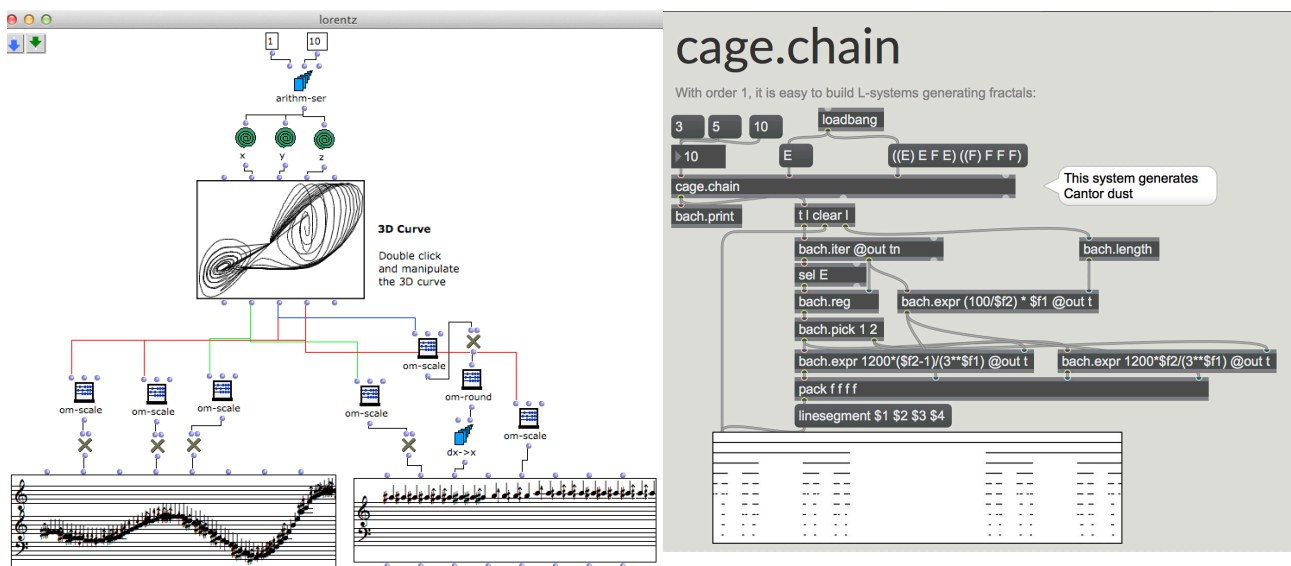
*Ecrits ou l'Invention de la musique spectrale*, édition établie par Guy Lelong, collection Répercussions, Paris : MF, 2008.

<sup>13</sup> Marco Stroppa, *Organismes d'information musicale : une approche de la composition*. La musique et les Sciences Cognitives, Contemporary Music Review, Pierre Mardaga, Bruxelles, 1989.

<sup>14</sup> Marco Stroppa, *Connaître des beautés insoupçonnées*. Entretiens du festival Présences. 2016.

environnements les plus connus étant OpenMusic<sup>15</sup>, Patchworks<sup>16</sup> et Common Music<sup>17</sup> tous les trois reposant sur le langage CommonLisp. Les langages par contraintes<sup>18</sup> permettent également d'exprimer, en les modélisant par des ensembles de contraintes, des pratiques comme le contrepoint, l'harmonie ou l'instrumentation. Plus récemment le développement de la librairie Bach et d'un langage dédié pour le logiciel Max/MSP, développé par Andrea Agostini et Daniele Ghisi, a permis d'ajouter un aspect interactif et « temps-réel » à l'univers de la composition assistée. Elle intègre des notations et modules de traitement de données musicales symboliques. Le package Cage<sup>19</sup> introduit notamment la notion de meta-information dans les objets musicaux de la partition : une note peut par exemple déclencher un processus « caché » supplémentaire lors de son exécution, comme une courbe d'automation.

Ces divers langages ont représenté des moyens de générer du matériau, d'élaborer des processus complexes et de générer des partitions (destinées à des instruments comme à des générateurs de synthèse). Ils sont aujourd'hui « augmentés » des travaux concernant les divers descripteurs du son et la classification — qui ont des implications dans l'aide à l'orchestration<sup>20</sup> mais également dans la gestion des sonothèques de compositeurs — ou encore les diverses recherches sur les réseaux de neurones.



À gauche l'implémentation en OpenMusic d'un attracteur de Lorenz<sup>21</sup>, à droite un patch Bach (bibliothèque Cage) implémentant un système de Lindenmayer.

15 Jean Bresson, Carlos Agon, and Gérard Assayag. *Visual lisp/clos programming in openmusic*. Higher-Order and Symbolic Computation, 22(1) :81–111, 2009.

16 Mikael Laurson, *PATCHWORK : a Graphic Language in PREFORM*. Ann Arbor, MI : MPublishing, University of Michigan Library, 1989.

17 Heinrich Taube, *Common music : A music composition language in common lisp and clos*. Computer Music Journal, pages 21–32, 1991.

18 Francesca Rossi, Peter van Beek et Toby Walsh, *Handbook of Constraint Programming*. Elsevier.

19 Andrea Agostini, Daniele Ghisi et Eric Maestri, *Recreating Gérard Grisey's Vortex Temporum with cage*, dans Proceedings of the International Computer Music Conference (ICMC), Utrecht. 2016

20 Voir les développements récents fait à l'Ircam sur le logiciel Orchids notamment.

21 Patch OpenMusic réalité par Jean-Michel Darrémont.

## Du paramétrique au relationnel

Les différents outils d'aide à la composition, si ils sont très efficaces pour générer, le sont assez peu pour penser une organisation du matériau au sens large. Nous avons vu qu'une notion revient régulièrement dans les différents modèles compositionnels exposés précédemment : les rapports entre local et global, chaque approche privilégiant généralement une orientation ou une autre (du local au global ou inversement).

Il nous apparaît que la prédominance des paradigmes paramétriques — privilégiant la génération de matériau par combinaison de paramètres ou par l'analyse d'un son donnant lieu à un rendu de données utilisables par la suite comme image à « recomposer » — a favorisé cette polarisation, établie au fil du temps, entre forme et matériau. Elle ne facilite pas aujourd'hui la conceptualisation de la complexité des rapports entre échelles locales et globales.

Forts de ce constat, il nous semble nécessaire d'opérer un changement de paradigme, passant ainsi d'une logique paramétrique et combinatoire à une « logique » relationnelle, plus à même de représenter les rapports *dialectiques* existants dans le domaine de la composition et notamment capable de procéder à des aller-retours entre échelles. Il ne s'agit donc pas, dans notre recherche, de créer des outils ou des représentations qui génèrent à proprement parler un matériau musical, mais bien davantage de trouver un moyen de penser son organisation, dans l'optique d'ouvrir des voies allant à double sens entre les différentes échelles temporelles qui irriguent une œuvre musicale. Non pas l'unique solution à un problème donc, mais, peut-être, l'émergence d'un chemin de traverse.

## La notion de *Denkraum*

Nous opérons ici un pas de côté de façon à pouvoir passer ensuite au cœur de notre propos avec, nous l'espérons, plus de clarté quant aux origines et aux motivations de notre recherche.

Le terme *Denkraum* appartient au vocabulaire d'Aby Warburg, historien de l'art dont le travail a notamment jeté les bases de l'iconologie. En 1919, à Hambourg, Warburg créa une bibliothèque très particulière dans laquelle les livres n'étaient ni classés chronologiquement, ni alphabétiquement mais selon un rapport « de bon voisinage » — « quand vous allez prendre un livre dans les rayons, celui dont vous avez réellement besoin n'est pas celui-là mais son voisin ». L'ordre de classement des livres, qu'il révisait chaque jour, devait, en toute vraisemblance, sembler bien chaotique pour un visiteur qui n'aurait pas été familier de ce que Warburg nommait son « espace de pensée » (*Denkraum*). Il s'agissait bien là — en plus de ce qui deviendra l'un des plus importants centres de recherche en histoire de l'art et de la culture — d'une véritable bibliothèque d'idées.

Warburg, à chaque étape de son travail, imaginait de nouvelles connexions entre les livres, impliquant alors de nouveaux réajustements dans la classification. Chaque avancée dans son système de pensée, chaque nouvelle idée portant sur l'inter-relation de faits l'amenait à regrouper les livres correspondants. La bibliothèque changeait suivant sa méthode de recherche et la variation de ses intérêts. Autrement dit, la loi de bon voisinage rendait les rayons de sa bibliothèque dynamiques et amenait de ce fait les chercheurs qui s'y égarèrent vers des pensées non-familiales, vers des terrains qu'ils n'auraient parfois pu envisager dans un autre contexte.

*L'espace de pensée, en tant que bibliothèque labyrinthique, remplit l'intervalle de contradiction entre ordre et désordre.*

Selon Fritz Saxl<sup>22</sup>, Warburg souhaitait d'une certaine manière empêcher que les choses soient techni-

<sup>22</sup> Fritz Saxl, *The History of Warburg's Library*, 1944, dans Gombrich, *Aby Warburg. An Intellectual Biography*, Londres : The Warburg Institute (2ème éd. 1986), pp. 325-338.

quement trop faciles pour qui souhaiterait être formé à cette « méthode ». Il y avait cette sorte « d'épreuve du labyrinthe »<sup>23</sup> indispensable pour lui au progrès de la recherche.

Les catégories de classement évoluaient donc avec le temps, à partir d'un ensemble de critères plutôt que d'un seul. Comme Lévi-Strauss l'a bien remarqué, « l'exigence de l'organisation est un besoin commun à l'art et à la science » et le principe d'ordre, « la taxinomie, qui est la mise en ordre par excellence, possède une éminente valeur esthétique »<sup>24</sup>.



Photo de la Kulturwissenschaftliche Bibliothek Warburg

Chez Walter Benjamin<sup>25</sup>, le *Denkraum* pourrait représenter le jeu discret entre ce qui est loin et ce qui est proche : le lointain et le proche, deux instances identiques d'un point de vue théorique, habitent l'espace de pensée et le rendent visible. « Le *Denkraum*, soit-il bibliothèque, planche à images ou collection de passages, fonctionne comme un espace-dynamite : il charge les bornes magnétiques de la pensée ». L'espace de pensée serait donc un lieu indéterminé d'un point de vue géographique, un espace fictionnel chargé d'interrogations sur l'histoire et sur la pensée, l'*endroit invisible* dans lequel l'*écoulement temporel* se rend visible. Le *Denkraum* benjaminien est l'espace tendu entre l'idée de temps et la représentation de cette idée. Cette représentation, énigmatique et métaphorique, constitue le cœur du *Denkraum* : en franchissant la frontière entre l'existence quotidienne et la conscience historique, l'être humain ouvre une porte donnant accès à une réflexion sur l'histoire.

Quel serait, dès lors, le *Denkraum* du compositeur, celui susceptible d'actualiser et de re-présenter les mouvements invisibles de sa pensée ?

« Le sens d'une question, c'est la méthode pour y répondre. [...] Dis-moi comment tu cherches, je te dirai ce que tu cherches. »<sup>26</sup>

Avec cette brève digression, nous souhaitons mettre en évidence plusieurs aspects qui nous semblent fondamentaux dans le travail compositionnel et qui irriguent la notion de *Denkraum*. Tout d'abord, il y a un principe de *distance critique* avec l'objet (l'histoire pour Warburg et — nous le verrons plus tard — l'espace compositionnel pour nous). Si l'espace de pensée warburgien est un lieu d'actualisation de la mémoire, il est « impossible de le parcourir sans être pris dans le filet de questions tissé par l'historien hambourgeois, impossible de rester à la surface des problèmes »<sup>27</sup>. Le procédé inventé par Warburg avec sa bibliothèque puis plus tard avec l'*Atlas Mnemosyne* n'est pas seulement une méthode heuristique, c'est une réification de la pensée relationnelle.

23 Voir ici l'article de Maud Hagelstein, *Mémoire et Denkraum. Réflexions épistémologiques sur la Kulturwissenschaftliche Bibliothek Warburg*, Conserveries mémorielles, #5 | -1, 38-46.

24 Claude Lévi-Strauss, *La pensée sauvage*, Paris : Plon, 1962, p. 21.

25 Walter Benjamin, *Images de pensée*, Paris : Christian Bourgeois Éditeur, 1998.

26 Ludwig Wittgenstein, *Tractatus logico-philosophicus*, Paris : Gallimard, 1975.

27 Maud Hagelstein, *Mémoire et Denkraum. Réflexions épistémologiques sur la Kulturwissenschaftliche Bibliothek Warburg*.

Si l'on voulait doucement se rapprocher de notre sujet, nous pourrions imaginer un espace — fût-il abstrait — dans lequel des objets musicaux (au sens morphologique) seraient placés et disposés en fonction de leurs affinités et des associations possibles entre chacun d'eux. Il serait alors question de pouvoir considérer une grande partie des relations qui existeraient entre ces objets et d'en avoir une représentation qui soit susceptible d'être exploitable, qui soit en mesure de *figurer* une certaine pensée musicale.

## Classification et montage : du Denkraum à l'Atlas

« Il invente, entre tout cela, des zones interstitielles d'exploration, des intervalles heuristiques. [...] Il est un outil, non pas de l'épuisement logique des possibilités données, mais de l'inépuisable ouverture aux possibles non encore donnés. Son principe, son moteur, n'est autre que l'imagination. »  
[...]

« L'atlas Warburgien est un objet pensé sur un pari. C'est le pari que les images, assemblées d'une certaine façon, nous offriraient la possibilité — ou, mieux, la ressource inépuisable — d'une relecture du monde. »<sup>28</sup>

Avec l'atlas *Mnémosyne*<sup>29</sup> (la photo ci-contre montre la planche 32), Aby Warburg a pu réaliser le dispositif que sa recherche « attendait » en quelque sorte depuis toujours : un médium lui permettant de manipuler comme des *objets interprétants*<sup>30</sup> les images mêmes qui constituaient d'abord ses *objets à interpréter*. « Avec *Mnémosyne*, Warburg fonde “une iconologie des intervalles” qui ne porte plus sur des objets, mais sur des tensions, des analogies, contrastes ou contradictions»<sup>31</sup>. Si nous nous intéressons ici à la *forme atlas* comme paradigme de pensée, c'est que celle-ci diffère de tout catalogue et même de toute archive supposée intégrale. L'imagination qu'il suscite permet d'y déceler de nouveaux rapports, de nouvelles correspondances et analogies qui seront elles-mêmes inépuisables « comme est inépuisable toute *pensée des relations* qu'un montage inédit sera susceptible de manifester ». La question du montage prend donc ici une place absolument fondamentale, dans le sens où la redistribution qu'il implique amène une relecture donnant une *lisibilité* à ce qui n'est *que visible*. Ce qui nous intéresse donc ici, c'est la faculté d'un procédé à susciter de nouvelles idées — en modifiant la disposition d'un *espace de pensée* — et par cela à soulever de nouvelles problématiques. Une partie du travail effectué pour cette thèse a consisté à trouver un moyen équivalent, d'un point de vue conceptuel, qui permette de rendre *lisible* un matériau qui pour les raisons explicitées précédemment est très souvent caractérisé par l'hétérogénéité des nombreuses informations qui le composent.

À une époque où le travail de la forme fait davantage place au déroulement linéaire des successions d'états et de sons étagés ou juxtaposés par le truchement de l'écoute « sur le vif », il est essentiel de créer des systèmes qui soient à l'image de l'atlas warburgien : « des lieux » dans lesquels la pensée musicale se construit à travers un tissu de relations complexes et interconnectées dynamiquement. Des lieux qui favorisent une certaine distance critique vis-à-vis des objets qui les composent. Il ne s'agit pas d'élaborer un modèle qui soit une « copie » de l'Atlas warburgien mais bien de construire une pensée qui s'en inspire d'un point de vue conceptuel.

Il ne s'agit donc pas là de trouver un outil de classification générique mais plutôt d'imaginer un modèle de représentation qui puisse tenir lieu d'espace de pensée. L'espace topologique que nous verrons au

28 Georges Didi-Huberman, *Atlas ou le gai savoir inquiet*, *L'Œil de l'histoire* 3, Paris : Les Éditions de Minuit, 2011, p. 11-79.

29 L'atlas *Mnémosyne* est un corpus d'images créé en 1921 et 1929, étroitement lié à la bibliothèque d'Aby Warburg. C'est une œuvre unique, dont l'ambition était de poser les fondements d'une grammaire figurative générale. Il s'agissait d'une nouvelle manière de comprendre les images à partir de leur juxtaposition. Cet atlas moderne juxtaposait, montait ensemble, des éléments hétérogènes pour « mettre l'histoire de l'art en mouvement(s) ».

30 *Ibid.*

31 Philippe-Alain Michaud, *Aby Warburg et l'image en mouvement*, Édition Macula, 1998, p. 238.

chapitre suivant est vu comme une immense table de montage et comme établi de la pensée musicale. Il s'agit d'y faire émerger des singularités, des rapports tellement particuliers qu'ils seront capables de déterminer l'identité musicale d'un espace, comme les spécificités d'un montage peuvent déterminer un certain mode de pensée. Plus généralement et nous y reviendrons, il s'agira d'essayer d'y voir émerger certaines qualités de temps pour parvenir à franchir l'orée de l'édification formelle.

*« C'est d'une connaissance traversière qu'elle nous fait don, par sa puissance intrinsèque de montage qui consiste à découvrir – là-même où elle refuse les liens suscités par les ressemblances obviees – des liens que l'observation directe est incapable de discerner. »<sup>32</sup>*



Planche 32 de *l'Atlas Mnemosyne*

32 Georges Didi-Huberman, 2011, p. 11-79.



# ESPACES COMPOSITIONNELS

« *Le paradigme est un cas singulier qui n'est isolé du contexte dont il fait partie que dans la mesure où, en présentant sa propre singularité, il rend intelligible un nouvel ensemble dont il constitue lui-même l'homogénéité.* »<sup>33</sup>

Lorsqu'on conçoit une entité musicale – un geste, une phrase, un germe – il faut mettre en œuvre un agencement d'éléments intrinsèquement différents mais qui forment, par leur mise-en-relation, une unité immanente à cet agencement. Il se construit un rapport complexe entre un tout (une phrase, un motif, un son...) et ses parties. Un agencement a des composantes potentiellement très hétérogènes (données temporelles, timbriques, fréquentielles, articulatoires, spatiales...). Il faut alors se demander comment tout ces éléments peuvent se conjuguer pour former une entité dont l'identité musicale soit suffisamment forte pour être distincte de ses voisines et comment la complexité d'une seule entité musicale (de taille plus ou moins grande) peut s'organiser avec d'autres et donc prendre en compte un certain contexte. Une des questions centrales, à l'origine de cette recherche, a été de savoir comment organiser, dans un même espace, les multiples paramètres aujourd'hui utilisés en composition afin de parvenir à « voir » le matériau de façon à ce que les potentialités formelles qu'il recèle soient en mesure d'apparaître et deviennent pour le compositeur un terrain d'exploration en lui-même. Il s'agissait avant tout de trouver notre propre *espace de pensée*.

Il est un problème récurrent dans les divers « systèmes » compositionnels et dont nous souhaiterions, sinon trouver une possible solution, au moins mettre en évidence la problématique. La constitution « d'unités musicales », ou de processus amenant une prolifération, une génération d'éléments le plus souvent de taille réduite amène souvent une construction séquentielle et s'occupe souvent exclusivement d'une organisation temporelle limitée à l'échelle des éléments avec lesquels elle travaille, à savoir de petites unités, non liées à une organisation beaucoup plus globale, celle qui se rapproche du travail formel. Il se produit alors une dichotomie entre une conception extrêmement travaillée, singularisée et « réfléchie » du point de vue du matériau et le travail de la forme, davantage issu d'une approximative « cuisine compositionnelle ».

Nous nous sommes donc intéressés à des structures particulières que l'on retrouve en topologie algébrique et qui nous semblent avoir une réelle cohérence musicale. Le parti pris de départ résidait alors dans un choix d'organisation de structures comportant de multiples paramètres mais ne se limitant pas à un paradigme purement combinatoire, de façon à approcher une pensée beaucoup plus relationnelle.

L'idée d'avoir des structures capables d'accueillir tous les paramètres dont nous avons besoin dans la composition — de figurer leurs liens, leurs rapports de voisinages, de connexité ou d'obstruction pour pouvoir par la suite imaginer des parcours, des sauts et, au-delà même du matériau, d'esquisser une forme — nous amena donc vers des structures topologiques. Il devait s'agir à la fois d'un lieu de création, d'organisation et d'exploration composé d'éléments exclusivement symboliques. Ces éléments, souvent interprétables de diverses manières, sont sensés incarner une certaine identité musicale et temporelle. L'enjeu était ici d'arriver à figurer les relations qui se tissent entre chacun des éléments constitutifs du matériau, pour pouvoir choisir des chemins en toute conscience, avec une appréhension temporelle plus élargie. Enfin, il était question de pouvoir manipuler des structures nécessairement complexes puisqu'ayant pour fondement l'union d'éléments hétérogènes et plus ou moins interdépendants.

<sup>33</sup> Georgio Agamben, *Signatura rerum*. Editions Vrin.

# Topologie algébrique

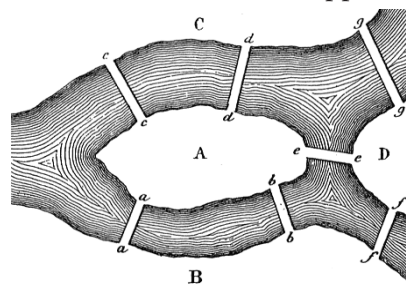
« La notion de passoire est indépendante de la notion de trou et réciproquement. »

Devise Shadock

Henri Poincaré parlait d'un type de géométrie<sup>34</sup> où la quantité (métrique) est complètement bannie et qui est de ce fait purement qualitative. Dans ce qu'il appelait l'*analysis situs*, « deux figures sont équivalentes toutes les fois qu'on peut passer de l'une à l'autre par une déformation continue. Un cercle est équivalent à une ellipse ou même à une courbe fermée quelconque »<sup>35</sup>. L'analyse des lieux consiste alors à voir comment ceux-ci sont articulés entre eux et comment passer de l'un à l'autre. L'articulation se fait via la connexion — qu'on peut associer historiquement à Euler et son problème des 7 ponts de Königsberg<sup>36</sup>. Mais au-delà des graphes, il s'agit avec la topologie d'avoir des morceaux d'espaces à plusieurs dimensions, qu'on pourra manipuler et sur lesquels on pourra faire des opérations (de l'algèbre). C'est donc une approche constructive.

La théorie des catégories<sup>37</sup> qui étudie les structures mathématiques et leurs relations développe un langage pour la topologie algébrique. La notion de voisinage est ici une notion primitive. L'algèbre a une approche très symbolique, très adaptée dans notre cas, pour construire, manipuler des objets et les transcrire sous forme de programmes informatiques.

Nous avons choisi, de manière inévitablement orientée, une représentation particulière de la topologie algébrique, celle des complexes simpliciaux.



## Musique et topologie

Il est arrivé qu'en théorie de la musique, on représente des familles d'objets ou de propriétés par des structures spatiales symboliques. La nature algébrique d'un grand nombre de notions musicales (tempérament, canons rythmiques, groupes d'intervalles, etc.) suggère en effet de les reformuler sous la forme d'espaces discrets ou continus dont l'étude des propriétés combinatoires, géométriques ou topologiques ont inspiré de nouvelles approches en théorie comme en analyse musicale.

C'est dans la seconde moitié du XVIII<sup>ème</sup> siècle que Euler propose de considérer les notes et les tonalités comme des points d'un espace bidimensionnel, une représentation géométrique qu'il appelle le « *speculum musicum* ». En opposition à la représentation circulaire, introduite par Marin Mersenne plus d'un siècle auparavant, Euler considère l'espace tonal comme engendré par deux axes représentant les intervalles à partir desquels tout intervalle peut être calculé. Grâce au développement de la théorie des groupes et, notamment, au théorème de décomposition de Sylow, on découvrira par la suite que les deux représentations sont de facto équivalentes — le groupe cyclique d'ordre 12 (c'est-à-dire le cercle) étant isomorphe en tant que groupe au tore en tant que produit des sous-groupes d'ordre 3 et 4 respectivement.

Des théoriciens ont par exemple recours à des *orbifolds* pour la représentation de progressions harmo-

34 Henri Poincaré, *Dernières pensées* (Bibliothèque Philos. scient.). Paris, Ernest Flammarion, 1913.

35 Maurice Fréchet et Ky Fan, *Introduction à la topologie combinatoire*. Paris, Éditions Jacques Gabay, 1946.

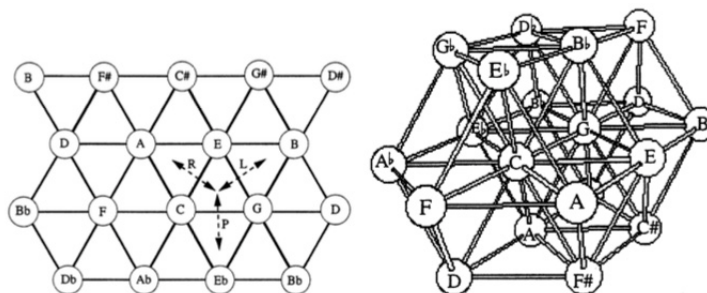
36 Ce problème est un problème mathématique à l'origine de la topologie et de la théorie des graphes, résolu par Euler en 1736. Il consiste à déterminer s'il existe un chemin dans les rues de Königsberg permettant de passer une et une seule fois par chaque pont et de revenir à son point de départ. Euler regarde alors ce problème comme un graphe.

37 La théorie des catégories est mise en place par Samuel Eilenberg et Saunders Mac Lane en 1942-1945, motivée par les besoins complexes de mises en relation de structures mathématiques dans le cadre de la topologie algébrique. Elle se diffuse dans les années 1960-1970 en France par Alexandre Grothendieck, qui en fit une étude systématique.

niques<sup>38</sup>, ou au graphe des hauteurs, plus connu sous le nom de *Tonnetz*, pour la représentation de transformations d'accords typiques de la seconde moitié du courant romantique, dites transformations néo-riemanniennes<sup>39</sup>.

Ces représentations se sont focalisées sur un paramètre uniquement : celui des hauteurs. Cela s'explique par la prédominance de celui-ci dans le langage tonal. Il s'agissait avant tout de visualiser les déplacements d'un accord à l'autre.

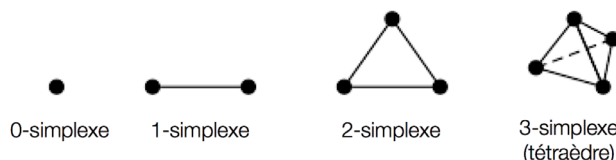
Notre démarche n'a aucun rapport avec ce type de représentation, même si nous avons adopté un modèle topologique. En effet, notre but était de pouvoir représenter l'intégralité des constituants du matériau utilisés pour la composition. En aucun cas il ne s'agissait de se focaliser sur un seul aspect, décorrélé du reste. Cette précision étant faite, nous pouvons désormais présenter notre modèle, en commençant par une introduction aux complexes simpliciaux.



A gauche un *Tonnetz* néo-Riemannien et à droite une dérivation tridimensionnelle du *Tonnetz* par E. Gollin.

## Complexe simplicial

Un complexe simplicial abstrait est un espace topologique construit par « recollement » de points, de segments de lignes, de triangles, ainsi que leurs homologues n-dimensionnels, les simplexes. Les simplexes sont les « briques » de base pour construire cet espace. Les faces d'un simplexe peuvent se comprendre comme les enveloppes convexes des sous-ensembles des sommets. Le terme « face » comprend donc ici les sommets, les arêtes, les faces triangulaires et ainsi de suite en dimension supérieure. La topologie algébrique introduit la notion de complexe simplicial pour formaliser la juxtaposition des n-simplexes dans un cadre algébrique et discret pouvant donc servir à l'implémentation d'outil sur un ordinateur.



L'intuition fondamentale qui nous guide dans cette représentation est que si deux choses interagissent, on peut considérer qu'elles sont voisines dans un certain espace. Les complexes simpliciaux permettent de modéliser des relations n-aires où n est supérieur ou égale à deux. Ils peuvent donc figurer des objets complexes, comme des unités musicales comportant un grand nombre de paramètres.

Ron Atkin a proposé l'utilisation des complexes pour la représentation de relations binaires notamment en sciences sociales et a introduit des outils d'analyse pour ces représentations : la Q analyse. Nous esquissons très rapidement ici comment la structure du complexe simplicial permet de représenter diverses sortes de relations, avant de présenter plus en détails notre utilisation de ses structures dans une optique d'organisation du matériau.

*L'intuition fondamentale qui nous guide dans cette représentation est que si deux choses interagissent, on peut considérer qu'elles sont voisines dans un certain espace.*

38 C. Callender, I. Quinn, D. Tymoczko. *Generalized voice-leading spaces*. Science, vol. 320, n°5874, 2008. p.346.

39 R. Cohn. *Neo-riemannian operations, parsimonious trichords, and their « Tonnetz » representations*. Journal of Music Theory, vol. 41, n°1, p.1-66.

## Q-analyse : la représentation de relations binaires

Au début des années 70, Ron Atkin a proposé de représenter une relation binaire  $R$  entre deux ensembles quelconques par un complexe simplicial : la Q analyse<sup>40</sup>. Les travaux autour de la Q analyse ont surtout porté sur les domaines de la modélisation des relations sociales<sup>41</sup>, l'analyse des interactions entre agents, l'analyse de trafic, la reconnaissance des formes, l'analyse des positions aux échecs, etc. Des extensions aux raisonnements diagrammatiques et aux logiques de description ont été développées à la fin des années 90<sup>42</sup>. La présentation qui suit est issue de la thèse d'Erika Valencia<sup>43</sup>.

### Relations binaires

Une relation binaire d'un ensemble  $A$  vers un ensemble  $B$  est une règle  $\lambda$  qui associe des éléments de  $B$  à des éléments de  $A$ . Pour déterminer complètement  $\lambda$  il suffit de connaître tous les paires d'éléments  $(a, b)$  où  $a$  appartient à  $A$  et  $b$  appartient à  $B$  qui vérifient le prédicat défini par la règle  $\lambda$ . Ainsi  $\lambda$  est entièrement définie par un sous ensemble de  $A \times B$ . La relation associée à  $\lambda$  et partant de  $B$  pour retourner dans  $A$  est appelée son *inverse* et est notée  $\lambda^{-1}$ .

Une relation peut être représentée à l'aide d'un tableau prenant des valeurs binaires, sa *matrice d'incidence*  $\Lambda$ . Prenons un exemple : soit  $A = \{a_1, a_2, a_3\}$  et  $B = \{b_1, b_2, b_3\}$ . On peut définir :

$$\lambda \subset (A \times B), \lambda = \{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$$

Alors  $\Lambda = \lambda_{ij}$  sera une matrice  $Card(A) \times Card(B)$ , donnée par :

$\lambda$	$b_1$	$b_2$	$b_3$
$a_1$	1	0	0
$a_2$	0	0	1
$a_3$	0	0	1

où  $\Lambda_{ij} = 1$  si l'élément  $(a_i, b_j) \in \lambda$  et 0 sinon.

Une fonction  $f$  est un cas particulier de relation pour laquelle il correspond au maximum un élément dans  $B$  à chaque élément de  $A$ . Le domaine de  $f$  est la restriction de  $A$  aux éléments  $a_i \in A$  pour lesquels  $f(a_i) \in B$ . Lorsque l'on a la propriété :

40 Ron Atkin, *Combinatorial Connectivities in Social Systems: an application of simplicial complex structures to the study of large organizations*. (Vol. 34). Birkhauser, 1977.

Ron Atkin, *Multidimensional Man: Can Man Live in 3-dimensional Space ?*. Penguin Press, 1981.

Johnson, J. H., & Loomes, M. J. *The mathematical revolution inspired by computing*. Clarendon Press, 1991.

41 *Ibid.*

Gould, P., *Q-analysis, or a language of structure: An introduction for social scientists, geographers and planners*. International Journal of Man-Machine Studies, 13(2), 1980. p169-199.

Casti, J., *Topological methods for social and behavioral systems*. International Journal Of General System, 8(4), 1982.

42 Valencia, E., Giavitto, J. L., & Sansonnet, J. P. (1998, August). *Algebraic Topology for Knowledge Representation in Analogy Solving*. In ECAI. p88-92.

Valencia, E., Giavitto, J. L., & Sansonnet, J. P. (1998). *Esqimo: Modelling analogy with topology*. In *Second European Conference on Cognitive Modelling (ECCM2)*, p212-213.

Giavitto, J. L., & Valencia, E. (2001). *Diagrammatic Representation and Reasoning, chapter A Topological Framework for Modeling Diagrammatic Reasoning Tasks*.

Valencia, Erika. *Outils de topologie algébrique pour la gestion de l'hétérogénéité sémantique entre agents dialogiques*. Thèse de doctorat de l'université de Paris Sud, 2000.

43 Valencia, Erika. 2000.

$$f(a_i) = f(a_j) \Rightarrow a_i = a_j$$

alors  $f$  est une *injection*. Lorsque, tous les éléments de  $B$  sont une image d'un élément de  $A$  par  $f$ , on parle de *surjection*. Lorsque l'on a à la fois une injection et une surjection, on dit que  $f$  est une bijection, ou encore que  $f$  est *bijective*.

Si une relation  $f$  est transitive, réflexive, symétrique alors elle est appelé *relation d'équivalence*, elle a pour effet de partitionner  $A$ . Une partition est une division de  $A$  en sous-ensembles d'intersections nulles et dont l'union redonne exactement  $A$ . Les sous-ensembles de  $A$  créés par partition sont les classe d'équivalence de  $\lambda$  sur  $A$ .

### Représentation des relations binaires par un complexe simplicial

La matrice  $\Lambda_{ij}$  que nous avons associée à  $\lambda$  ne peut correspondre qu'à une relation binaire. Nous allons nous intéresser à ces relations binaires, spécifiées sous forme de matrice, et voir quel est leur lien avec les complexes simpliciaux.

Supposons que  $\Lambda$  contienne  $m$  lignes et  $n$  colonnes, c'est à dire que  $Card(A) = m, Card(B) = n$ . Fixons  $B$  comme l'ensemble des sommets, ainsi un ensemble de  $(p+1)$  éléments de  $B$  formera un  $p$ -simplexe. Prenons maintenant les éléments de  $A$  qui sont impliqués dans la relation  $\lambda$ . Soit  $a_i$  l'un d'entre eux. Il est alors possible qu'il apparaisse à plusieurs endroits dans l'ensemble de définition de  $\lambda$ , par exemple :

$$\lambda = \{ \dots, (a_i, b_j), \dots, (a_i, b_k), \dots, (a_i, b_l), \dots \}$$

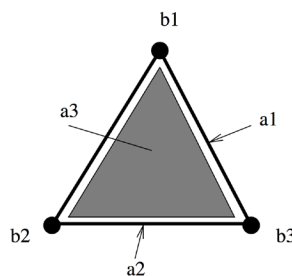
On peut dire que  $a_i$  est  $\lambda$ -relié à  $b_j, b_k, b_l$ . Ce fait sera visible sur la matrice d'incidence comme l'ensemble de tous les 1 de la ligne correspondant à  $a_i$ . L'ensemble  $b_j, b_k, b_l$  des éléments liés à  $a_i$  est inclus dans  $B$ , ce sont donc des sommets. Ainsi, on peut définir un simplexe associé à  $a_i$  et qui prend appui sur trois sommets,  $b_j, b_k, b_l$ . Chaque ligne de la matrice définit un simplexe, l'ensemble des ces simplexes étant caractéristique de  $\lambda$ , on dit alors qu'ils forment le complexe simplicial associé à  $\lambda$ . En prenant par exemple pour  $\lambda$  :

$$\lambda = \{(a_1, b_1), (a_1, b_3), (a_2, b_2), (a_2, b_3), (a_3, b_1), (a_3, b_2), (a_3, b_3)\}$$

alors nous obtenons la matrice d'incidence  $\Lambda$  associée à  $\lambda$  :

$\lambda$	$b_1$	$b_2$	$b_3$
$a_1$	1	0	1
$a_2$	0	1	1
$a_3$	1	1	1

On peut représenter  $a_3$  par un 2-simplexe défini par les trois sommets :  $a_3 = \langle b_1, b_2, b_3 \rangle$  puisque  $a_3$  est en  $\lambda$ -relation avec ces trois sommets. Nous pouvons donc représenter la matrice d'incidence sous forme de polyèdres : le complexe simplicial associé à  $\lambda$  est le suivant :



## Représentation générale de relations n-aires

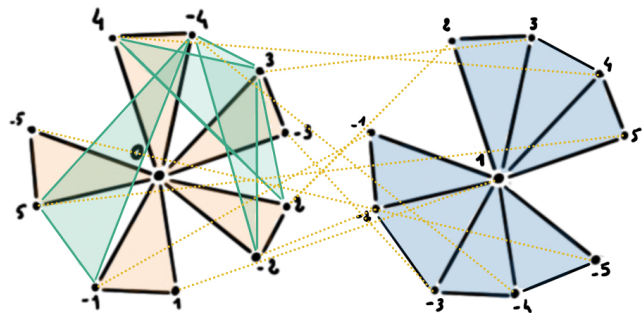
Il est naturel d'étendre la représentation topologique des représentations binaires  $R(a_1, a_2)$  aux relations n-aires en général  $R(a_1, a_2, \dots, a_n)$ . Pour simplifier, nous supposons que les relations considérées sont indépendantes de l'ordre des éléments : autrement dit, les relations binaires sont commutatives et pour une relation n-aire générale, nous écrivons  $R\{a_1, a_2, \dots, a_n\}$  pour exprimer cette indépendance<sup>44</sup>.

La représentation des relations n-aires diffère dans son principe de la représentation des relations binaires. Dans le cas binaire, un p-simplexe  $\{a_1, a_2, \dots, a_p\}$  est créé s'il existe un  $b$  tel que  $R(b, a_1)$  et  $R(b, a_2)$  et ... et  $R(b, a_p)$ . Dans le cas n-aire, on crée un simplexe  $\{a_1, a_2, \dots, a_p\}$  s'il existe  $b_1, b_2, \dots, b_{n-p}$  tel que  $R\{a_1, a_2, \dots, a_p, b_1, b_2, \dots, b_{n-p}\}$  (pour une relation qui serait dépendante de l'ordre des éléments il faudrait un entrelacement quelconque des  $a_i$  et des  $b_j$ ). L'information représentée par le simplexe est donc différente dans les deux cas : dans le premier, un simplexe est associé à un élément  $a$  et représente des images de  $a$  par  $R$ . Dans le cas général, un simplexe représente (partiellement) une occurrence de la relation  $R$ . On peut noter par exemple que la dimension maximale d'un simplexe associé à une représentation binaire est le nombre maximal d'images d'un élément (c'est donc un nombre arbitraire) alors que pour une représentation n-aire, la dimension maximale d'un simplexe est  $n-1$ .

Dans les deux cas, les simplexes associés à la relation  $R$  possèdent la propriété caractéristique d'un complexe : si un simplexe  $\{a_1, \dots, a_k\}$  est associé à  $R$ , alors tout sous-ensemble de  $\{a_1, \dots, a_k\}$  est aussi un simplexe associé à  $R$ .

Un exemple permet d'illustrer cette construction. Prenons  $R$  une relation ternaire<sup>45</sup> définie par  $R(a, b, c)$  si et seulement si  $a = b.(c-1)$ . Sur l'ensemble  $E = \{-5, -4, \dots, -1, 1, 0, 1, \dots, 5\}$   $R$  est associé aux simplexes

- $\{q, -q, 0\}$  pour  $q \in E-\{0\}$  car  $q = -q.(0 - 1)$
- $\{q, 1, q+1\}$  pour  $q \in E-\{1, 5\}$  car  $q = 1.(q+1 - 1)$
- $\{q, -1, -q+1\}$  pour  $q \in E-\{-1, -5\}$  car  $q = -1.(-q+1-1)$
- $\{4, 2, 3\}$  car  $4 = 2.(3-1)$
- $\{-4, -2, 3\}$  car  $-4 = -2.(3-1)$
- $\{-4, -1, 5\}$  car  $-1 = -1.(5-1)$



## Représentation spatiale et organisation du matériau compositionnel

Pour simplifier, le simplexe de dimension 3 que nous utiliserons formalise la co-existence de 4 types de propriétés musicales. Le choix des 4 éléments est inhérent aux nécessités compositionnelles qui nous sont ici propres et que nous précisons par la suite. Si nous avons choisi ce modèle plutôt que celui d'une clique dans un graphe, c'est qu'il était nécessaire qu'un simplexe formalisant une sorte « d'entité musicale » ne puisse pas changer de dimension, dans une même représentation. Autrement dit, chaque entité musicale, pour être définie, nécessite 4 sommets liés entre eux et formant de ce fait un 3-simplexe. Si nous avons choisi d'utiliser un modèle de graphe, nous aurions obtenu un graphe exclusivement constitué de

<sup>44</sup> Cette hypothèse ne vise qu'à simplifier les écritures ci-dessous.

<sup>45</sup>  $R$  n'est pas indépendante de l'ordre des arguments et on adapte la construction comme indiquée.

cliques ce qui d'un point de vue conceptuel nous paraissait moins pertinent.

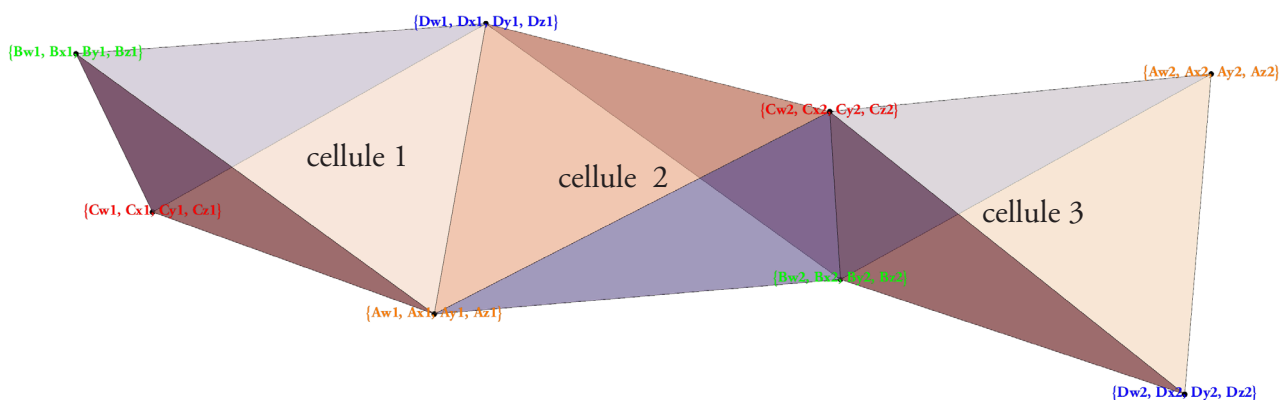
Dans notre contexte, aucun logiciel permettant de créer, de visualiser et de manipuler aisément ces types de structures n'existe encore aujourd'hui. Nous avons donc, pour l'implémentation, utilisé le logiciel de calcul formel Mathematica pour parvenir, de façon encore tout à fait laborieuse, à créer ces espaces. L'implémentation demeure encore aujourd'hui relativement rudimentaire et mériterait un gros travail de développement qui n'a pu être qu'initié durant cette thèse.

## Méthode de construction d'un espace

*« J'aimerais qu'il existe des lieux stables, immobiles, intangibles, intouchés et presque intouchables, immuables, enracinés ; des lieux qui seraient des références, des points de départ, des sources. »<sup>46</sup>*

Nous détaillons ici les étapes de construction de nos espaces compositionnels<sup>47</sup>. Il s'agit au départ d'entrer, de manière séquentielle, le matériau qui sera utilisé pour une (ou plusieurs) pièce afin d'en avoir une représentation spatiale qui serve de base de travail à la composition. La première étape réside donc dans l'énumération des divers sommets (les 0-simplexes de notre espace) qui seront de 4 types et de ce fait, lorsqu'ils seront réunis par la suite, formeront nos « cellules élémentaires », les 3-simplexes. Ces différents types comportent, nous verrons pourquoi par la suite, 4 paramètres, qui par conséquent peuvent également être représentés sous forme de 3-simplexes. Nous verrons que cela permet, à partir d'un même espace, d'obtenir 4 sous-espaces (limités à un seul type de propriété musicale chacun) résultants des 4 différents types de sommets du complexe simplicial.

L'illustration suivante montre 3 cellules (3-simplexes) composées chacune par 4 types de « qualités musicales » — représentées ici pour simplifier notre explication par les lettres A, B, C et D — formant les différents sommets. Ces sommets sont eux-mêmes composés de 4 paramètres (ici w, x, y et z) qui peuvent eux-mêmes être représentés sous forme de 3-simplexe.



Il s'agit donc d'intégrer incrémentalement à l'espace compositionnel les cellules avec lesquelles nous souhaitons travailler. Celles-ci sont déclarées de façon tout à fait indépendante les unes des autres. La fonction implémentée dans Mathematica les assemble par la suite en fonction des sommets qu'elles ont en commun. Dans la figure ci-dessus, les cellules 1 et 2 ont les sommets D et A pour éléments communs et sont donc reliées.

<sup>46</sup> Georges Perec, *Espèces d'espaces*. Éditions Galilée, 2000.

<sup>47</sup> Nous renvoyons le lecteur aux annexes si il souhaite voir plus précisément l'implémentation de la représentation.

## Qu'est-ce qu'une cellule ?

Dans son acception usuelle, une cellule est généralement définie comme l'unité morphologique ou fonctionnelle d'un ensemble organisé. Plus petite unité vivante capable de se reproduire de façon autonome, une cellule capable de se différencier en plusieurs types de cellules est appelée pluripotente<sup>48</sup> en biologie. Chez la plupart des organismes multicellulaires, toutes les cellules ne sont pas identiques. Elles présentent des différences importantes au niveau de leur morphologie et de leur fonction. Cependant, tous les différents types cellulaires sont dérivés d'une seule cellule-œuf fécondée et ce, grâce à la différenciation. Nous arrêterons ici notre analogie pour nous concentrer sur ce que représente *musicalement* cette cellule dans notre espace compositionnel.

Si celle-ci est souvent comprise en musique comme le plus petit élément — rythmique, mélodique ou harmonique, apparaissant de manière isolée ou comme partie d'un tout plus étendu — nous lui préférons une définition beaucoup moins circonscrite. Notre cellule — définie par un ensemble de  $4 \times 4$  paramètres réifié en un 3-simplexe — est autonome et interprétable. Elle est sensible à son environnement, à savoir dans notre cas au contexte musical dans lequel elle s'inscrit. Sa temporalité (nous reviendrons sur cette question dans un chapitre dédié) dépend des paramètres mis en relation au sein de sa structure et des cellules qui l'environnent. Une cellule doit avoir, par le rassemblement des 16 paramètres qui la composent, une identité musicale forte.

## Composantes paramétriques

Les paramètres choisis pour l'actuelle version de notre représentation spatiale sont symboliques et peuvent être modifiés, dans leur nature même, par le compositeur. La représentation ne serait en aucun cas affectées pas un changement de paramètres. La nature et les composantes paramétriques ont évolué au cours de notre recherche et des diverses compositions. Nous présentons ici la version qui a servi à la composition de la pièce *Namenlosen* écrite en 2017. L'espace compositionnel de cette pièce avait été partiellement utilisé pour la pièce *Sortir du noir* écrite l'année précédente. Nous détaillons par la suite chaque paramètre par souci de clarté mais nous soulignons ici l'importance — bien plus grande nous semble-t-il — du principe même de mise en relation incarnée par la structure du simplexe.

La composition des différentes cellules est la suivante<sup>49</sup> (la lettre correspondante dans l'espace est indiquée entre parenthèses) :

- Simplexe fréquentiel (f) :
  - matrice d'intervalles
  - niveau de polarisation
  - profil
  - ambitus
- Simplexe rythmique (r) :
  - métrique
  - vitesse
  - profil
  - régularité
- Simplexe d'attraction (a) :
  - symétrie harmonique
  - symétrie rythmique
  - tension timbrique
  - directivité
- Simplexe timbrique (t) :
  - enveloppe
  - dynamique générale
  - timbre principal
  - timbre secondaire (complété par un degré d'hétérogénéité)

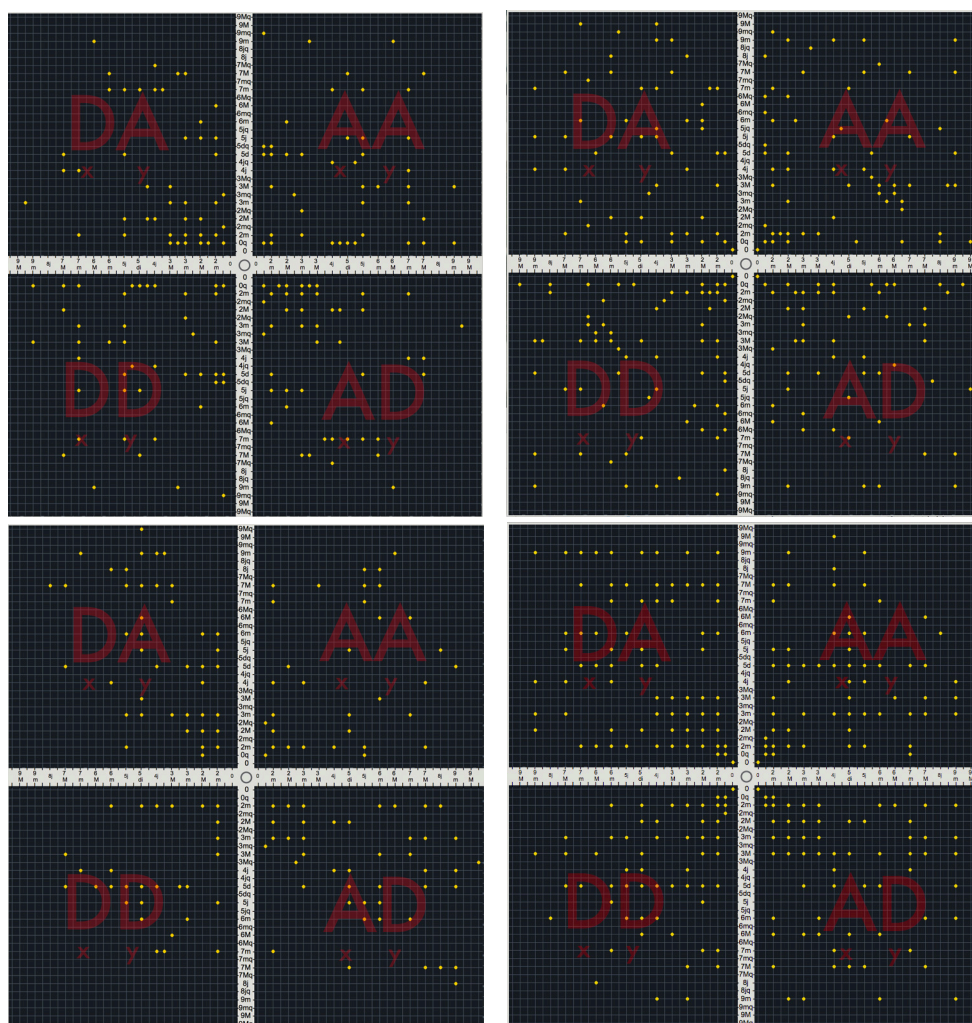
<sup>48</sup> Ces cellules sont appelées cellules souches chez les animaux et cellules méristématiques chez les plantes supérieures.

<sup>49</sup> Nous appuyons ici sur le fait que cette famille de paramètres n'est en aucun cas une sorte de liste exhaustive des paramètres nécessaires à la composition aujourd'hui. Il en existe d'autres (on pense notamment aux nombreux paramètres des actuels descripteurs du son), qui par ailleurs pourraient tout à fait être intégrés à notre présent modèle par la suite. Il nous paraît tout à fait fructueux d'avoir une représentation qui puisse accueillir des paramètres de toutes sortes et n'en soit pas pour autant altérée, permettant ainsi à ce modèle de représentation d'évoluer au cours du temps.

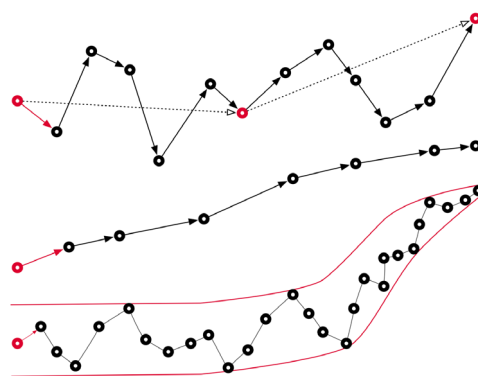


**Les simplexes fréquentiels** concernent tout ce qui implique les hauteurs. La matrice d'intervalle définit les successions permises d'un intervalle à l'autre, en différenciant les ascendants et les descendants. La couleur harmonique se perçoit alors dans le temps. Le niveau de contrainte peut augmenter si les successions s'appliquent non seulement aux intervalles horizontaux (au sens mélodique) mais également verticaux (harmonique).

Les quatre matrices présentées ci-dessous sont celles utilisées pour la composition de *Namenlosen*. Les lettres qu'on peut discerner à l'intérieur (DA, AA, DD, AD) spécifient le sens des intervalles : descendant → ascendant, ascendant → ascendant, descendant → descendant et ascendant → descendant.

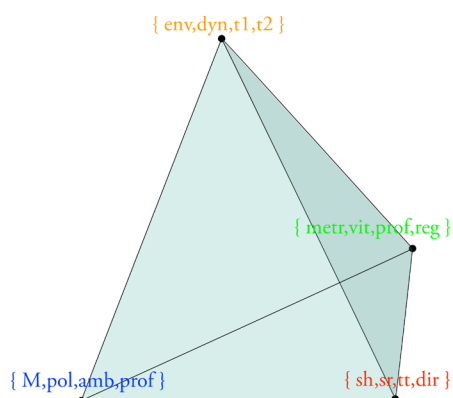


Plusieurs fonctions développées en Antescofo<sup>50</sup> permettent de générer des séquences de hauteurs en fonction de certaines contraintes. En spécifiant une hauteur et un intervalle de départ, l'algorithme génère une séquence de  $n$  intervalles. On peut spécifier des hauteurs obligatoires à l'intérieur de la séquence (voir la première ligne du schéma de droite comme des hauteurs placées sur un axe de temps en abscisse et dont les éléments imposés par le compositeur sont en rouge) qui forment ainsi des points pivots en début, milieu et fin de séquence. On peut imposer un type d'intervalles (ascendant ou descendant) sur toute la sé-



<sup>50</sup> Voir la partie « Organiser le temps » de cette thèse pour plus de détails.

quence. Dans le schéma, la 2ème ligne est une séquence qui n'a généré que des intervalles ascendant. On peut également spécifier un ambitus à l'intérieur duquel la séquence doit s'inscrire comme dans la 3ème ligne du schéma. Ici l'algorithme génère des séquences qui se situe dans un ambitus qui évolue dans le temps. Le niveau de polarisation indique quant à lui le degré de fixation sur une seule hauteur.



**Les simplexes rythmiques** intègrent une métrique, utilisée de façon assez souple comme moyen de division d'une cellule (dans un contexte d'organisation d'une phrase) ou comme métrique au sens plus classique (organisation de la pulsation à une échelle très locale). Le profil indique les accélérations, décélérations, stagnations (et différentes combinaisons) de la vitesse moyenne quand la régularité donne le type de striage de la cellule.

**Les simplexes timbriques** intègrent des notions liées au timbre et à l'articulation. On y trouve une spécification d'une enveloppe générale qui s'applique sur la cellule concernée mais peut éventuellement s'appliquer de façon plus locale. Cette enveloppe est associée à une dynamique générale. Un timbre principal et un timbre secondaire sont associés. Il peut s'agir d'un type de timbre enrichi en harmonique (*e*), mat (*m*), naturel (*n*), résonnant (*r*), saturé (*s*) ou bruité (*b*). Les combinaisons de timbre permettent une spécification plus précise. A cette combinaison est ajouté un degré d'hétérogénéité (facultatif) déterminant l'hétérogénéité des timbres qui seront utilisés dans la cellule.

**Les simplexes d'attraction** rassemblent des paramètres liés à des questions de directivité et de statisme. La conjonction de symétries harmoniques et rythmiques plus ou moins grandes, d'un degré de tension timbrique et d'une directivité permet de se placer sur un large spectre allant d'un état totalement statique à un autre très directif et pointant vers un ou plusieurs pivots temporels. Ce type de simplexe permet d'attribuer à une cellule une *direction* plus ou moins grande. Cette spécificité sera déterminante lorsque nous aborderons la question du temps.

Les simplexes *f*, *r*, *t* et *a* sont représentés dans l'espace compositionnel par un seul sommet chacun, réunissant les 4 paramètres (voir illustration page suivante).

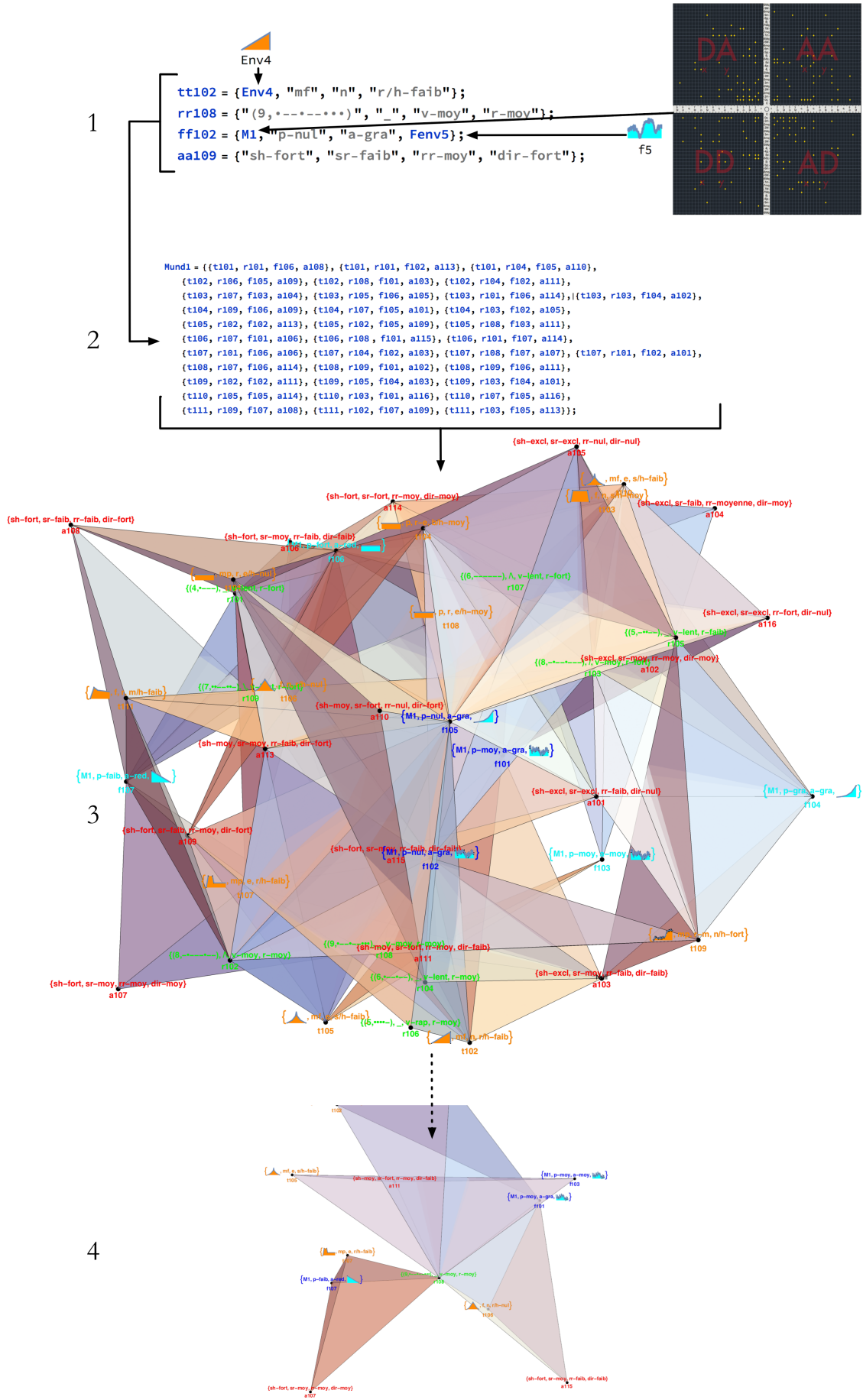
### *Du 0-simplexe au complexe simplicial*

Le schéma de la page suivante montre la constitution d'un espace. La première étape consiste à entrer des simplexes (avec leurs 4 paramètres respectifs). Il s'agit *a priori* d'entrer l'intégralité du matériau d'une pièce. Mais nous pourrions ajouter n'importe quel type de simplexe à tout moment par la suite, selon les nécessités compositionnelles. Après cela, nous pouvons créer les premières cellules en assemblant 4 simplexes (étape 2). De la même manière que précédemment, nous pourrions ajouter autant de cellules que nécessaire.

À partir de cette base de cellules, un complexe simplicial est alors créé en fonction des sommets, arêtes et faces communes, pour former notre espace compositionnel (étape 3). Deux cellules qui ont pour élément commun un sommet se retrouvent alors "collées" et réunies par ce sommet. Une fois l'espace compositionnel construit, on peut visualiser les rapports de voisinage de l'ensemble du matériau intégré à notre espace.

L'inconvénient de la représentation est sa complexité au sens graphique et de ce fait son manque de lisibilité. Pour remédier à ce problème une fonction permet de sélectionner un sommet, une arête, une face ou une cellule entière en particulier, pour voir uniquement ses voisins et filtrer le reste (étape 4).

# ORGANISER LE MULTIPLE : ESPACES COMPOSITIONNELS

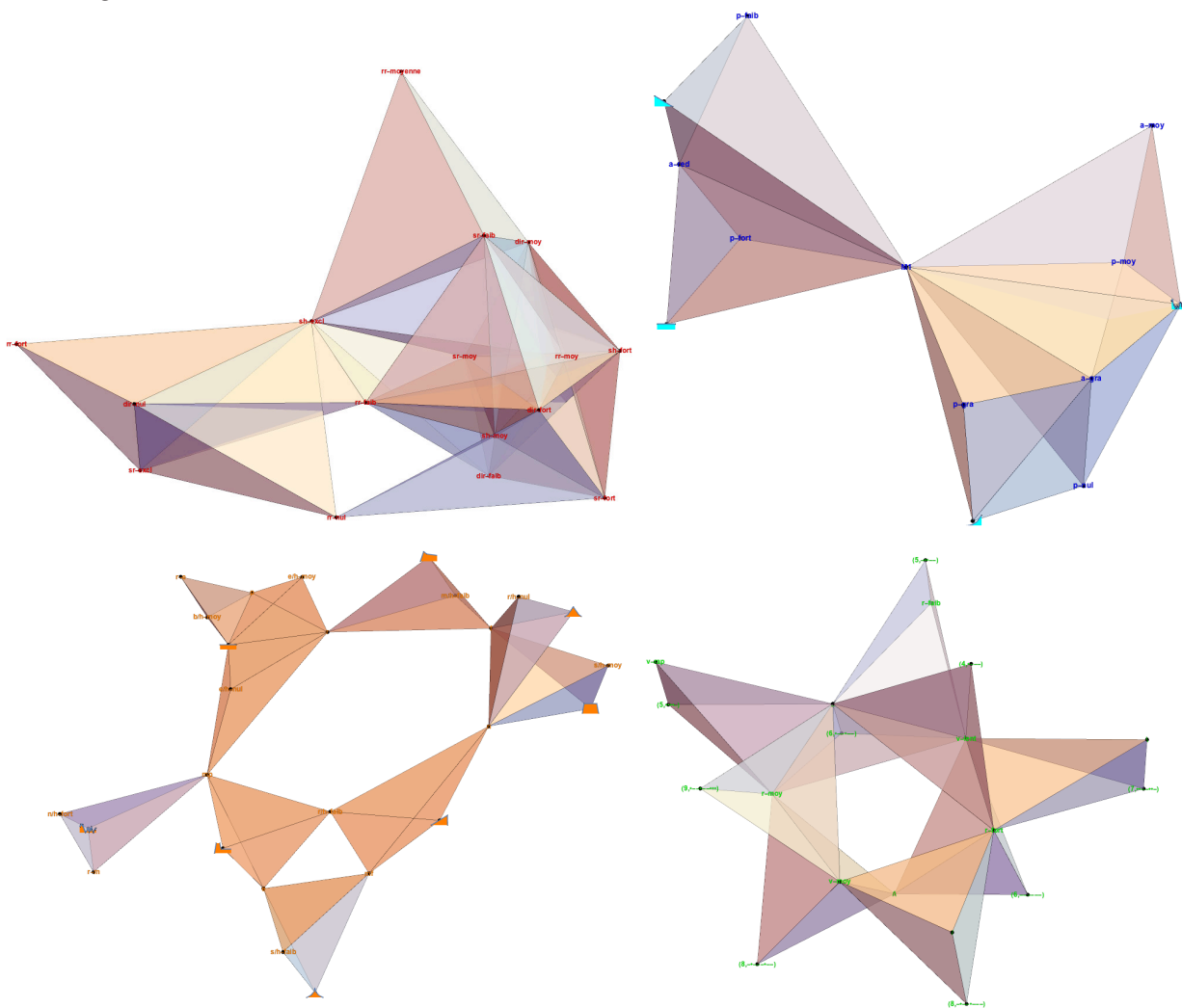


*Quatre sous-espaces*

La représentation topologique permet de visualiser un matériau contenant une quantité importante d'informations (nous l'avons vu, une cellule se compose de 4 simplexes contenant 4 éléments chacun). Mais elle sert également à explorer ce matériau « spatialisé » en choisissant le domaine par lequel on y entre (celui du timbre, des fréquences au sens large, du domaine rythmique ou des attractions). Un algorithme écrit en Mathematica permet de sélectionner directement un sous-espace particulier et de visualiser les éléments de l'espace général qui l'ont pour élément commun. On peut donc filtrer la visualisation selon la focalisation souhaitée.

L'espace n'est alors composé que de simplexes d'un même type et ce qui dans la représentation principale du complexe simplicial n'est représenté que par un sommet (un 0-simplexe composé d'une liste de 4 éléments), est alors représenté sous forme d'un 3-simplexe de 4 sommets. On peut alors choisir de polariser son attention sur un aspect particulier (le timbre par exemple) pour se construire une première image mentale d'un objet musical, sans perdre de vue les autres aspects du discours, sans cesse liés au pôle travaillé. Mais surtout sans quitter des yeux l'horizon du territoire à venir, en gardant en mémoire à la fois le chemin parcouru, et celui à trouver.

Les quatre figures suivantes représentent les 4 sous-espaces issus de l'espace compositionnel de la page précédente. En rouge l'espace d'attractions, en bleu l'espace fréquentiel, en orange l'espace timbrique et en vert l'espace rythmique. On peut ainsi passer d'un espace à l'autre, par aller-retours, en variant les types de voisinages.

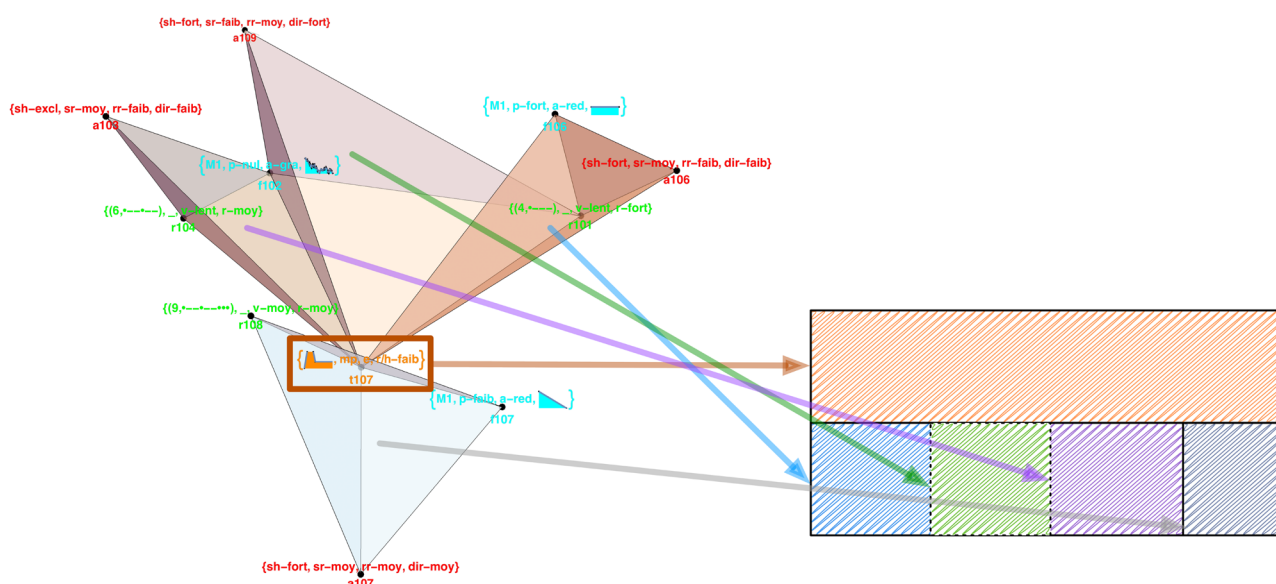


## Du local au global

Une cellule n'a pas de durée définie au préalable. Son interprétation peut donc se situer à diverses échelles, locales ou globales. Une même cellule peut servir pour définir les composantes d'un motif très bref, d'une longue phrase, voir d'un passage entier. Il nous semble essentiel aujourd'hui de pouvoir non seulement travailler à des échelles différentes, mais aussi d'imaginer des processus qui, définis à une certaine échelle, aient des implications, de par leurs composantes et leurs relations, à des échelles supérieures.

Dans la figure ci-dessous, nous avons filtré un espace compositionnel pour ne garder que les cellules dont l'invariant est "t107" (un simplexe timbrique défini par une attaque au départ suivie d'une longue période de relâchement, une dynamique générale *mp* et des timbres riches en harmoniques et résonnants). Nous avons ici 4 cellules distinctes avec 1 sommet commun (t107) à toutes et dont certaines ont deux éléments communs. Le simplexe timbrique va être ici utilisé sur une longue période durant laquelle nous passerons d'une cellule à l'autre avec de plus ou moins grands rapports de proximité. En passant de la première cellule (bleu sur le schéma) à la seconde, les paramètres rythmiques sont conservés. De la seconde (verte) à la troisième cellule (violette), c'est cette fois les paramètres fréquentiels qui restent inchangés.

On voit ici qu'un même simplexe peut être associé à des échelles très locales ou plus globales. Le temps se construit non pas par des paramètres rythmiques locaux mais par les relations entre divers paramètres évoluant eux-mêmes sur une certaine durée<sup>51</sup>.

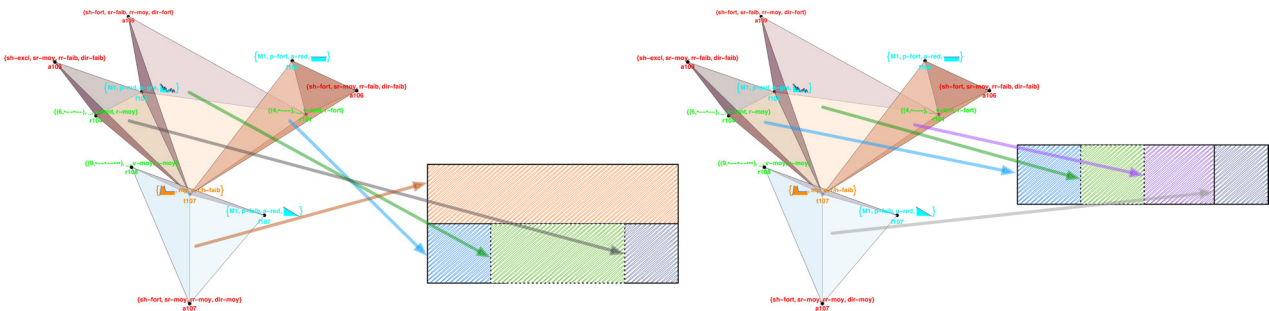


Nous proposons une interprétation du schéma présenté ci-dessus comme une possibilité parmi d'autres. Les quatre mesures correspondent aux quatre cellules (en bleu, vert, violet et gris sur le schéma).



51 Voir le chapitre sur la dialectique des temps.

Les figures ci-dessous montrent 2 autres utilisations des mêmes cellules. La figure de gauche superpose une cellule à une séquence de 3 cellules. Celle de droite est une utilisation très classique, passant d'une cellule à l'autre selon une plus ou moins grande proximité (1 ou 2 sommets communs).



## Parle-t-on de distance ?

Nous avons pris le parti de ne pas tenir compte des distances lors du calcul des positions des sommets dans la structure. Ce calcul est assuré par Mathematica à partir d'algorithmes de placement de graphes. Ce faisant, la notion de distance peut parfois être trompeuse dans la représentation, étant donné le fait que le « contenu d'un sommet » n'est pas analysé. Ainsi, un sommet peut être relativement éloigné d'un autre alors même que leurs identités sont proches. Cela est dû à notre position de départ, à savoir qu'une entité musicale n'est identifiée qu'à partir du moment où elle est représentée par un 3-simplexe (une cellule). De ce fait, un sommet pris individuellement ne peut faire sens à lui tout seul. C'est la co-existence et les relations entre tous les sommets du simplexe qui induisent une identité musicale propre à celui-ci.

On peut donc parler de distance topologique, mais elle se distingue nécessairement de la notion de distance qu'on trouve par exemple dans le graphe utilisé pour classer les descripteurs en traitement du signal<sup>52</sup>. On ne peut en aucun cas « fixer » de quelconques valeurs sur un axe ou un repère orthonormé permettant de les comparer en termes *quantitatifs*. Ce qui nous importait, dans le choix d'un mode de représentation et d'organisation, c'est qu'il puisse exprimer des choses en termes *qualitatifs*. L'aspect symbolique est donc ici fondamental.

## La nécessité du symbole

Les données intégrées aux espaces utilisés sont exclusivement symboliques, permettant plusieurs interprétations durant la composition. Une même cellule pourra produire des résultats différents selon l'interprétation qui en aura été faite au moment de l'écriture de la partition ou de la partie électronique. Cela permet de rendre les structures « agnostiques », à savoir qu'elles peuvent parler d'une écriture purement instrumentale, électronique ou mixte. Ce modèle ne génère pas un matériau instantanément et cela pour cette raison fondamentale : il est interprétable. Cette approche va à l'encontre de celles qui favorisent une objectivation des qualités musicales ou une génération par apprentissage profond<sup>53</sup>. La notion de *qualité musicale* est ici une *construction* propre au compositeur. Nous ne nous exprimons pas avec les termes habituellement utilisés pour décrire un flux audio, notamment dans les descripteurs<sup>54</sup> audio et qui, souvent excessivement nombreux, peuvent s'avérer très peu explicites d'un point de vue purement musical. Il s'agit

<sup>52</sup> On citera notamment la représentation en 2 dimensions du logiciel de synthèse concaténative par corpus Catart. Voir l'article de Diemo Schwarz. *Interacting with a Corpus of Sounds*. eContact! online journal of the Canadian Electroacoustic Community, vol. 2, n° 16.

<sup>53</sup> Jean-Pierre Briot, Gaétant Hadjeres, François Pachet, *Deep learning techniques for music generation — A survey*. Arxiv:1709.01 620, septembre 2017.

<sup>54</sup> voir l'article de G. Peeters, B. Giordano, P. Susini, N. Misdariis, St. McAdams. *The Timbre Toolbox: Audio descriptors*

davantage de termes compositionnels pouvant s'appliquer, nous l'avons vu, à diverses échelles.

Bon nombre d'outils et de techniques compositionnelles donnent lieu à des générations de données « brutes » destinées à être directement retravaillées dans les esquisses et la partition. La pièce est alors composée à partir de ce matériau retravaillé, développé ou amplifié de diverses manières. Les interfaces utilisées fréquemment dans ces cas là se situent souvent dans le domaine de l'exploration ; un certain nombre de données sont générés et le compositeur doit alors « choisir », parmi la profusion de possibilités données par le système, les éléments qu'il utilisera par la suite dans la composition. C'est quelque part la technique du « pêcheur » qui place sa ligne, mouline et recommence jusqu'à ce que le bon poisson, tout du moins celui qu'il considère comme « recevable », se présente. En choisissant tel ou tel descripteur — tel ou tel appât si nous voulons filer la métaphore — il pourra alors biaiser, autant que faire se peut, le résultat et « orienter la prise ».

Notre perspective de travail est ici tout à fait différente puisqu'elle s'inscrit non pas tellement dans une démarche de génération d'un matériau mais comme moyen d'organiser et de construire une *pensée musicale relationnelle*. Ce mode de représentation oblige d'une certaine manière le compositeur à ne pas penser en terme uni-dimensionnel, à ne pas être focalisé sur un seul paramètre mais au contraire à chercher sans cesse à penser en terme d'ensemble, de relations dans une même cellule mais également de relations *dans le temps*. De ce fait, elle tente de se situer à la lisière du matériau et de la conception formelle. Les modes de représentation graphique ont toujours fortement orienté la musique et influencé la création musicale. Nous pensons qu'un renouvellement du mode de représentation du matériau musical est fondamental pour pouvoir réfléchir à un renouvellement des formes.

Notre traitement du timbre est exemplaire de cette approche. La détermination d'un timbre n'est pas ici définie par un ensemble de partiels ou de descripteurs mais par des termes suffisamment généraux pour être applicables à tous type d'instruments, acoustiques ou électroniques et suffisamment précis pour discriminer plusieurs timbres entre eux. Une enveloppe de type attaque-résonnance peut s'interpréter de bien des manières selon le type d'instrument ou d'ensemble à laquelle elle s'applique. Elle pourra également avoir des interprétations différentes selon le temps dans lequel elle s'inscrit. Cependant, on saura aisément discriminer, quelque soit l'interprétation concrète qui en est faite, une enveloppe de ce type avec une autre de type crescendo-decrescendo (▲). Cet aspect symbolique permet donc une grande souplesse et une expressivité du langage bien plus grande que si nous devions manipuler des listes de paramètres sous forme numérique. La spécificité d'une cellule, la précision de sa définition ne se joue donc pas dans celle d'un paramètre éventuellement manipulable mais, encore une fois, dans la réunion de plusieurs.

Si l'on souhaitait rendre la représentation « générative », il nous suffirait d'associer des paramètres précis à chaque paramètre symbolique et faire ainsi des tables de correspondance. L'aspect symbolique permet donc également des utilisations plus « classiques » et n'exclue pas une approche générative.

*« Le symbole a pour caractère de n'être jamais tout à fait arbitraire ; il n'est pas vide, il y a un rudiment de lien naturel entre le signifiant et le signifié. »<sup>55</sup>*

---

*of musical signals*. JASA, 130 (5) 2011

55 Ferdinand de Saussure, *Cours de linguistique générale*. Éditions Bally et Sechehaye, (1971). 1916.

## Parcours et directivité : la relation temps/espace

Toute représentation, et d'autant plus lorsqu'il s'agit d'un *espace de pensée*, est un choix signifiant. Celle que nous avons choisi n'expose aucune forme de temps classique. Il n'y a pas d'axe sur lequel comparer plusieurs valeurs et les situer dans le temps comme il n'y a, *a priori*, pas d'avant ou d'après, sinon l'évolution et la plus ou moins grande expansion de notre espace. La position d'un sommet dans l'espace compositionnel n'a de sens que dans la mesure où il est lié aux autres. Nous pourrions envisager cette représentation comme une sorte de carte dynamique qui évolue et se modifie au cours du temps. Une carte à plusieurs dimensions à partir de laquelle on peut envisager divers parcours, divers montages possibles en ayant bien en tête la « géographie » des lieux.

Si l'on pense à une organisation spatiale du matériau, se pose donc très rapidement le problème de la représentation, mentale et réelle, du temps dans celle-ci. Ce problème se pose fréquemment dans le traitement du matériau lorsque des paramètres temporels n'y sont pas intégrés. La question temporelle se trouve alors repoussée à plus tard et devient souvent exclusivement réservée au domaine formel. On peut légitimement se demander ce que signifie une intégration de données temporelles dans des structures spatiales. C'est que nous nous attachons à une conception du temps encore une fois plus qualitative que quantitative et qu'en ce sens, nous pouvons imaginer des organisations qui nous placent, d'une certaine manière, *devant le temps*.

La question temporelle est irréductible aux seuls paramètres classiquement admis comme étant les attributs du temps musical (tempo, vitesse, durées ...etc.). Il apparaît aujourd'hui évident que la composition d'un temps musical implique la prise en compte d'un ensemble de paramètres, de leurs évolutions et des corrélations pouvant exister entre chacun d'eux. On pensera notamment aux fameux « degrés de changement » chez Stockhausen qui mettent en lumière l'importance d'une prise en compte de chaque unité musicale (quelque soit sa nature : hauteur, rythme, timbre...etc) dans son contexte, horizontal et vertical. L'identification d'une qualité de temps musical s'établit alors à travers l'ensemble des paramètres musicaux et la notion d'intervalle<sup>56</sup>, le passage d'un état à un autre à une certaine vitesse, devient déterminante. Si nous avons intégré à nos espaces compositionnels des structures dont les paramètres sont uniquement temporels (ou plutôt rythmiques devrait-on dire), leur validité musicale n'est donc acquise qu'à partir du moment où elles sont liées aux autres éléments dont la spécification et l'organisation assure l'identité singulière des cellules musicales.

Deux principales questions surgissent immédiatement sur le rapport entre le temps et les données spatiales qui régissent nos structures topologiques. La première est de savoir comment l'intégration de données temporelles au sein du matériau peut s'accorder au passage à des échelles supérieures et notamment à l'échelle formelle. La seconde interroge l'existence d'un lien entre la notion de parcours dans un espace et celle de directivité formelle, de construction d'un discours musical. Le passage continu d'un point A à un point B assure-t-il pour autant une direction musicale, geste intentionnel d'une « source » vers son « but » ? De quelle manière associer ou dissocier le temps du parcours, inhérent à la topologie d'un espace, et les temps potentiellement multiples qui peuvent se jouer lorsqu'une forme se construit ?

Deux notions nous semblent cruciales, dans les réponses que l'on peut apporter à ces questions : la notion d'échelles et la notion de directivité. La question de l'échelle — locale ou plus ou moins globale — à laquelle on se place est, encore ici, déterminante. La notion de directivité implique une vue « à long terme » qui anticipe, à l'aune de chaque individualité passée, de nouveaux points de fuite, de nouvelles trajectoires. Pour pouvoir parler de directivité, il faut pouvoir envisager plusieurs temporalités en pa-

<sup>56</sup> Pour Zimmermann, l'intervalle est une notion centrale et fondatrice, de par la capacité de notre audition à percevoir la distance qui sépare deux sons. Et l'intervalle est perçu sous deux formes temporelles différentes, la succession et la simultanéité. (*Intervall und Zeit*).



rallèle, sur plusieurs échelles voire sur différentes dimensions (par exemple comment évolue tel timbre de tel endroit à tel autre et que se passe-t-il en parallèle du côté rythmique). C'est un « feuilletage de temps » qui se met en place à l'intérieur d'un même espace. Les passages d'une cellule à l'autre, par voisinage ou par saut, entraînent une multiplicité des temps, nous y reviendrons par la suite. Il reste toujours à voir comment l'impulsion expressive de l'instant peut se rattacher au courant plus profond.

## Des espaces autres

*« L'ordre, c'est à la fois ce qui se donne dans les choses comme leur loi intérieure, le réseau secret selon lequel elles se regardent en quelque sorte les unes les autres et ce qui n'existe qu'à travers la grille d'un regard, d'une attention, d'un langage; et c'est seulement dans les cases blanches de ce quadrillage qu'il se manifeste en profondeur comme déjà là, attendant en silence le moment d'être énoncé. »<sup>57</sup>*

Nous venons de le voir, un élément musical — qu'il s'agisse d'une note, d'un timbre, d'un rythme, quelque soit même sa nature — ne fait sens qu'à partir du moment où il interagit avec ce qui l'entoure et où sa co-existence avec d'autres éléments participe à la constitution d'un discours musical. De la même manière, un son pris en lui-même, avec toutes ses composantes, n'a de sens qu'à partir du moment où il est intégré à un ensemble, dans un certain « espace-temps » (dimensions verticales, horizontales et diagonales). La topologie nous semble représenter un intéressant outil conceptuel capable justement de « penser » la musique de manière relationnelle.

L'unité de base des structures que nous utilisons, la cellule (3-simplexe), réifie d'une certaine manière la réunion de paramètres hétérogènes en propriétés musicales définies symboliquement et constitue par cela un *potentiel* d'identité musicale interprétable. Cette unité de base co-existe avec d'autres unités formant alors un complexe simplicial. Celui-ci représente l'espace (topologique) compositionnel d'une pièce musicale, « lieu » de travail plus ou moins temporaire — puisqu'il peut être utilisé pour plusieurs pièces — où les notions de voisinage, de parcours, de proximité et d'obstruction peuvent être tout à fait fructueuses. La structure topologique devient lieu de conception, le matériau s'y déploie par palier à chaque ajout d'une unité musicale interprétable, d'une cellule. La conception d'un espace se fait de façon incrémentale : une cellule après l'autre. Le compositeur déclare au départ quelles sont les unités qu'il souhaite intégrer à l'espace de travail que représente le complexe simplicial en construction. Celui-ci devient espace de pensée, tout du moins *pour un certain temps*, construit en fonction des relations existantes entre chaque cellule déclarée.

Notre espace compositionnel, organisé en structures topologiques, comporte un certain nombre de particularités « géographiques ». En le parcourant, on peut distinguer des régions dans lesquelles des groupes de cellules partagent un certain nombre d'éléments communs et forment de ce fait une zone particulière. En partant d'un élément invariant et en se déplaçant d'une cellule à l'autre, on approche d'un principe assez classique de variation, avec par exemple un passage où les caractéristiques fréquentielles et rythmiques resteraient inchangées quand le timbre ou la directivité varierait dans le temps. En sautant d'une cellule à une autre sans préservation d'un élément commun, il serait davantage question de rupture, suivant le degré d'éloignement des qualités musicales d'une cellule à une autre. Suivre un long chemin allant d'une cellule à une autre par voisinage direct amènerait en revanche une écriture très continue dans son processus de développement. Cette continuité serait cependant tributaire du *degré de changement*<sup>58</sup>

<sup>57</sup> Michel Foucault, *Les mots et les choses*, vol. 166. Tel, Gallimard, 1966.

<sup>58</sup> Au sens où l'entend Stockhausen qui a repris cette notion issue de la théorie de l'information notamment dans *Carré*, pour quatre chœur et quatre orchestres mais aussi dans des pièces comme *Kontakte* où ces degrés de changement sont quasi-

entre chaque cellule. Garder un ou plusieurs invariants ne garantit pas forcément de continuité si les éléments changeants le sont de façon radicale. On peut se demander si ces spécificités n'entraînent pas avec elles la figuration de potentialités formelles (au sens musical du terme) propre à cet espace et si de ce fait, l'élaboration du matériau ne vient pas se confondre à l'édification formelle.

Les règles, les descriptions timbriques, temporelles, fréquentielles ou d'attraction sont autant de jalons qui marquent les « frontières ». Ils rythment la constitution de notre territoire, cet ensemble de matières d'expression qui se développent en « motifs territoriaux ». Pour tenter de dépasser la disjonction historique entre forme et matériau, faut-il alors penser que l'édification d'un « territoire compositionnel » implique, dans ses « souterrains », l'éruption de parcours, de sauts manifestant l'existence de devenirs formels ? Comme dans le paradoxe de l'œuf et de la poule, on peut poser la question de savoir si l'espace compositionnel précède la forme ou si c'est la forme, présente de façon plus ou moins consciente chez le compositeur au moment où il compose qui fait l'espace compositionnel. Or, il nous semble que nous nous positionnons sur le seuil de ces deux manières d'appréhender « le monde », quitte à tracer une troisième voie. À la fois nous entrons un matériau avec, disons en arrière-pensée, une *direction* vers laquelle nous nous dirigeons, une identité particulière que nous souhaitons en quelques sortes imposer à un espace. Et à la fois c'est en voyageant dans notre espace que la forme se construit petit-à-petit et que — en prenant conscience des relations existantes entre chaque constituant de l'espace/matériau — nous parvenons à *modeler* la forme en devenir. Les processus de construction, d'adaptation puis de remodelage du matériau que représente et organise notre espace compositionnel impliquent de ce fait une remise en cause de la forme.

En pareille situation le matériau ne serait plus alors ce réceptacle à matière neutre, résultat de calculs plus ou moins complexes et trop souvent désincarné par le fait même qu'il serait considéré comme se suffisant à lui-même, surdéterminant éventuellement la forme. Si la singularisation du matériau passe par la manière dont il est exploité pendant la composition, alors son renouvellement et sa « reconfiguration » peut venir de son articulation à la forme et de sa figuration au sein d'un territoire innervé par les trajectoires qu'il sous-entend et dont les *fonctions dramatiques* naîtraient des *rapports dialectiques* existants déjà en son sein.

« ... Alors le bateau cesse d'être boîte, habitat, objet possédé ; il devient œil voyageur, frôleur d'infinis ; il produit sans cesse des départs. »<sup>59</sup>

Les exemples sont nombreux, notamment dans l'histoire de la musique récente, de matériaux préfigurant, par leurs spécificités intrinsèques, des devenirs formels. Reprenons quelques références données en début de partie. Nous pourrions citer la notion de processus traitée bien entendu différemment chez Iannis Xenakis ou chez Gérard Grisey pour lequel celui-ci est un moyen d'assurer une progression vers le futur de l'œuvre<sup>60</sup>. Mais dans son cas, il est à noter que le processus exclut, d'un certain point de vue, les objets susceptibles de « s'imprimer » en mémoire. Seule la mémoire immédiate est sollicitée dans la mesure où c'est l'intrusion d'événements inattendus, de surcroît relativement rares, qui permet d'articuler une forme par processus. Il s'agit d'aller d'un état stable à un autre, en intégrant d'éventuels accidents<sup>61</sup>. De ce fait la forme reste relativement simple, le processus se confondant avec une partie ou la totalité d'une œuvre. Ici le matériau surdétermine la forme et le rapport entre les

*Le renouvellement du matériau viendrait alors de son articulation à la forme et de sa figuration au sein d'un territoire innervé par les trajectoires qu'il sous-entend et dont les fonctions dramatiques naîtraient des rapports dialectiques existants déjà en son sein.*

ment théorisés.

59 Roland Barthes, *Mythologies*, § « Nautilus » et « Bateau ivre ». Editions du Seuil, 1970.

60 Gérard Baillel, *Gérard Grisey, fondements d'une écriture*. Editions l'Harmattan, 2000.

61 À noter sur cette question l'intérêt porté aux travaux de René Thom par des compositeurs comme Hugues Dufourt.

deux est dirigé unilatéralement. Chez Boulez, deux concepts dominant la pensée<sup>62</sup> : la prolifération à partir d'une donnée initiale et la déduction des structures les unes par rapport aux autres. Avec ces deux concepts, le compositeur évite le recours au plan, la forme se construit dans l'instant et se ramifie à mesure que le compositeur continue de s'y investir (on connaît l'importance dans ce travail de la notion de *work in progress*). On pourrait pratiquement dire ici que le matériau génère la forme et que comme chez Grisey et même si la méthode est radicalement différente, le matériau surdétermine la forme. Le rapport à la mémoire est cependant chez Boulez assez différent puisqu'on y trouve un certain nombre de gestes « leitmotivs » (on pense ici, pour ne citer qu'elle, à *Sur incises*). Le compositeur a par ailleurs souvent fait référence au fameux réflexe de Pavlov. Chez Stockhausen, micro et macrostructure répondent aux mêmes lois, élaborées conjointement. C'est l'idée d'une totalité organique inscrite dans chaque partie constituante de l'œuvre, dans chaque « moment ». Il n'y a plus alors de tension entre forme globale et éléments singuliers. Contrairement aux exemples précédents ce n'est pas un mouvement de progression mais plutôt d'expansion qu'incarne ici la pensée compositionnelle — on connaît l'importance pour le compositeur de l'image de la spirale. Un même modèle s'incarne donc à la fois dans la forme et dans le matériau, sur diverses échelles. C'est une vision relativement transcendante de la composition. Nous l'avons vu, il existe beaucoup d'autres exemples plus récents dans lesquels les compositeurs se sont construits des sortes de modèles qui pré-déterminent de manière plus ou moins importante la forme. On citera notamment les exemples pré-cités au début de cette partie : le principe de grammaires génératives chez Philippe Manoury, d'organismes d'information musicale chez Marco Stroppa ou encore des différents modèles de fractales et de systèmes de Lindenmayer chez Alberto Posadas.

Face à un espace compositionnel souvent complexe, riches d'un nombre important de relations, il y a un spectre de réactions se situant entre deux pôles. Le premier est la réaction qui consiste à plonger — quitte à s'y noyer — dans cet espace de possibilités, à s'engouffrer à l'intérieur et à passer d'une cellule à l'autre, au hasard des trouvailles.

*« On ne quadrille pas un tel espace. Il faut donc prendre le risque de plonger. Plonger là-dedans, empathiquement, comme on plonge dans un océan sans limites connues, et se retrouver au plus profond des eaux. [...] Émettre cette comparaison, c'est laisser entendre que Warburg fut un chercheur d'un tout autre genre que ce "détective" ou ce "chasseur de tête" sous les traits duquel on a si souvent coutume de le figurer : il fut plutôt, dirai-je, un chercheur du genre pêcheur de perles. »<sup>63</sup>*

Le second pôle résiderait dans une réaction plus analytique et synthétique qui consisterait à percevoir les unités dans l'ensemble plus vaste du contexte général, à avoir un regard plus large et plus soucieux du contexte et du « com-prendre » — dans le sens de « prendre ensemble ». Au milieu de ces deux pôles se trouverait une position dans laquelle on cherche un élément particulier — et *déjà pressenti* — dans un espace rempli d'une profusion d'autres et qui, dans son principe, s'avère assez similaire à l'exemple donné par Jean-Paul Sartre :

*« J'ai rendez-vous avec Pierre à quatre heures. [...] "J'ai tout de suite vu qu'il n'était pas là"... Il est certain que le café, par soi-même, avec ses consommateurs, ses tables, ses banquettes, ses glaces, sa lumière, son atmosphère enfumée, et les bruits de voix, de soucoupes heurtées, de pas qui le remplissent, est un plein d'être. Et toutes les intuitions de détail que je puis avoir sont remplies par ces odeurs, ces sons, ces couleurs... Mais il faut observer que, dans la perception, il y a toujours constitution d'une forme sur un fond. Aucun objet, aucun groupe d'objets n'est spécialement désigné pour s'organiser en fond ou en forme : tout dépend de la direction de mon attention. Lorsque j'entre dans le café, pour y chercher Pierre, il se fait une organisation synthétique de tous les objets du café en fond sur quoi*

62 Philippe Albera, *Le son et le sens, Essais sur la musique de notre temps*. Editions Contrechamps, 2007.

63 Georges Didi-Huberman, *L'image survivante*. Les Éditions de Minuit, 2002, p. 507.

*Pierre est donné comme devant paraître... Chaque élément de la pièce, personne, table, chaise, tente de s'isoler, de s'enlever sur le fond constitué par la totalité des autres objets et retombe dans l'indifférenciation de ce fond, il se dilue dans ce fond. Car le fond est ce qui n'est vu que par surcroît, ce qui est l'objet d'une attention purement marginale. [...] Je suis témoin de l'évanouissement successif de tous les objets que je regarde, en particulier des visages, qui me retiennent un instant ("Si c'était Pierre?") et qui se décomposent aussi précisément parce qu'ils "ne sont pas" le visage de Pierre. Si, toutefois, je découvrais enfin Pierre, mon intuition serait remplie par un élément solide, je serais soudain fasciné par son visage et tout le café s'organiserait autour de lui, en présence discrète. »<sup>64</sup>*

Nous pourrions aussi parler des deux pôles représentant habituellement ceux des différentes manières de composer la musique que sont l'induction et la déduction. Ici, au sein de notre espace compositionnel, l'intégralité du spectre qui se situe entre ces deux pôles est continu. La démarche compositionnelle ne se situe jamais sur un point fixe pour ne plus y bouger. Ce sont généralement des allers-retours qui ont lieu durant tout le processus de composition et qui font du processus compositionnel un *processus dialectique*.

Les paradigmes que nous utilisons pour penser la musique, pour l'interpréter, doivent être en mesure d'assimiler toute la complexité des rapports qui existent à l'intérieur du matériau mais également l'ensemble des réactions que peut avoir le compositeur vis-à-vis de ce matériau. Ils doivent pouvoir refléter la totalité des spectres dont nous avons parlé. Dans ce contexte, il nous semble primordial de poser la question de savoir comment un matériau devient *singulier* et quel médium est en mesure de favoriser cette singularisation.

Nous pensons qu'une des voies possibles de cette singularisation réside dans une explicitation des relations qui lient chaque constituant du phénomène musical. Les éléments à prendre en compte sont extrêmement nombreux et c'est sur ce point particulier que la technologie, si elle est maîtrisée, peut nous permettre d'appréhender tous les niveaux de profondeur qui déterminent le devenir d'un matériau musical. Construire et expérimenter une organisation à partir de structures spatiales multi-dimensionnelles nous paraît extrêmement fertile pour cela. Le niveau de développement reste cependant très rudimentaire, ce qui limite la souplesse d'utilisation nécessaire à la composition. Notre version actuelle permet toutefois de visualiser puis de filtrer notre espace compositionnel, nous verrons à travers des exemples concrets comment nous l'avons utilisé pour les pièces réalisées pendant la recherche.

Choisir un modèle topologique comme représentation, comme moyen d'organisation et de génération du matériau n'est pas anodin. Ce modèle particulier nous amène à penser qu'il permet, intrinsèquement, une très forte singularisation due à sa caractéristique première : la notion de mise en relation, de voisinage. Décider que deux choses marchent ensemble, que deux éléments font partie d'une même entité ou qu'au contraire ils n'ont pas d'espace commun, délimiter des zones détenant certaines propriétés ou imaginer des rapports d'influence entre diverses cellules font éminemment partie du travail quotidien du compositeur.

Dans une époque où les outils « clef-en-main » fleurissent de toutes parts — où les interfaces sensées « rendre service à l'utilisateur en lui facilitant la vie » lui imposent en réalité une manière de penser et de fonctionner — il nous semble important de poser cette question de la singularisation du matériau et donc, comme on l'a vu précédemment, potentiellement de la forme. Le choix d'un environnement compositionnel ouvert et symbolique est une prise de position artistique forte, justement parce qu'il détermine un territoire, une géographie et un ensemble de conditions qui, sans imposer un seul et unique chemin, sont déjà la figuration d'une empreinte compositionnelle. Il nous appartient ici de créer des structures suffisamment ouvertes dans leurs définitions pour qu'elles puissent soutenir des pensées musicales singulières.

<sup>64</sup> Jean-Paul Sartre, *L'Être et le Néant : Essai d'ontologie phénoménologique*. Paris, Gallimard, coll. « Tel », 1976, p. 44.

Le travail du compositeur consiste souvent à trouver un certain « ordre des choses », à constituer un atlas protéiforme dans lequel cohabitent une faune extrêmement variées d'éléments qui se regardent en quelque sorte les uns les autres. Il doit trouver un moyen d'organiser l'amoncellement d'objets complexes avec lesquels il travaille. Il se peut que le compositeur établisse des relations entre des choses apparemment impossibles à réunir parce qu'il a le pressentiment qu'une unité de sens très précise est là, sous-jacente, qu'il la cherche et ne la découvre qu'au cours de son travail, en obéissant purement et simplement aux réflexes de l'intuition, cette prescience du devenir si nécessaire à l'imagination. Il est concevable que l'émergence de ces liaisons — qui peu à peu façonnent un discours musical propre au compositeur — puisse s'incarner dans des espaces tels que ceux que nous avons présenté. L'espace de pensée musicale, les processus compositionnels se présentent comme des intrications — de champs, de sens, de temps. « Si l'on veut regarder ce qui se passe à l'intérieur d'une intrication, si l'on veut connaître l'intimité vivante d'un tas de serpents, on découvre que le mouvement des corps emmêlés dessine tout un réseau d'intervalles eux-mêmes en mouvements »<sup>65</sup>. Comment s'embrassent et se séparent, comment s'opposent et se mêlent, comment s'éloignent et s'échangent les éléments d'une intrication, voilà bien les questions qui doivent habiter le compositeur lorsqu'il est face à son espace.

Penser l'entre-deux autant que les extrémités, prendre en compte cette polyphonie d'agencements qui se joue toujours davantage au sein du matériau et tenter d'y inscrire la grille d'une intention musicale, d'un langage à travers lequel cet enchevêtrement d'interdépendances prenne sens et s'entremêle aux possibles formels. Autrement dit, adopter une pensée de l'intervalle.

*« L'intervalle, c'est ce qui fait le temps impur, troué, multiple, résiduel. [...] C'est le contretemps, le grain de différence dans le rouage des répétitions. [...] C'est ce qui intrique et sépare alternativement les fils — ou les serpents — de l'écheveau du temps. C'est le chemin que parcourt une empreinte vers son incarnation. »<sup>66</sup>*

## De l'imagination des formes

*« L'idée est un pari : pari sur la validité, sur la détermination expressive, sur le potentiel de développement ; bref, pari sur quelques quantités plus ou moins connues, et sur nombre de qualités totalement inconnues. L'idée est un pari dans le sens où elle lie un certain nombre d'éléments sans savoir encore si ces éléments ne sont efficaces que dans cet amalgame de l'instant, ou s'ils sont susceptibles, en se développant l'un par rapport à l'autre, d'engendrer des conséquences multiples. »<sup>67</sup>*

Une idée naît parfois en dehors d'un quelconque modèle, se présentant comme un geste acoustique plus ou moins élaboré, un « geste d'intuition » dont la concrétisation ne peut quelquefois s'inscrire que dans un espace restreint. Cette « figure sauvage » vient parfois d'une réflexion, d'un cheminement au long court, souterrain, provenant parfois d'expériences antérieures diverses. Mais de cet objet initial, de cet hypothétique organisme en devenir, il faut révéler les possibilités, trouver les développements et savoir si l'on pourra ou non le ceindre dans un réseau de déduction ou le faire proliférer anarchiquement. Bien des outils existent aujourd'hui pour aider à « déduire » de ce type d'idées un matériau manipulable et développable (OpenMusic, PWGL, Bach, Opusmodus... pour ne citer qu'eux). Une idée peut d'ailleurs simplement venir de la manipulation d'un outil. On voit venir aujourd'hui, avec la profusion de recherches

<sup>65</sup> Georges Didi-Huberman, 2002, p. 500-501.

<sup>66</sup> *Ibid.*

<sup>67</sup> Pierre Boulez, *Leçons de musique. Points de repère III*. Éditions Christian Bourgeois, 2005.

sur l'apprentissage profond et les réseaux de neurones, des générateurs — puisant dans de gigantesques corpus — sensés engendrer du matériau *ad infinitum*. Les recherches musicales sur le matériau en lui-même n'ont jamais été si importantes et prolifiques que durant les cinquante dernières années.

Mais à partir de cela peut-on penser une forme globale ? Il est des formes dont l'invention découle de solutions toutes provisoires trouvées « sur le chemin », se découvrant au fur et à mesure du travail. La forme n'est jamais un déroulement prévisible de possibles épuisés à partir d'une réserve de taille limitée. Mais elle n'est pas non plus une suite de moments sans *relations sensibles*<sup>68</sup>. L'œuvre peut se concevoir dès l'origine, comme architecture, grande structure dont on discerne la trajectoire, les évolutions, parfois certaines caractéristiques plus précises, mais dont les composantes réelles ne sont pas encore imaginées. C'est pour donner corps à cette structure que les idées musicales devront surgir, parfois provenant d'idées restées inexploitées, comme latentes dans un atlas imaginaire et trouvant là le cadre approprié à leur développement. On voit que si l'écriture suscite inévitablement la forme, la forme peut aussi susciter l'écriture elle-même lorsque cela lui est nécessaire.

Dans nos espaces compositionnels, nous pouvons partir d'une idée et à partir de celle-ci créer une cellule qui lui corresponde puis essayer de développer celle-ci en multipliant au fur et à mesure les cellules qui viendront l'envelopper. Nous pouvons tout au contraire, à partir d'un espace constitué, chercher la cellule qui correspond à une idée restée trop vague et mal définie. L'écriture se plaçant ici sur un spectre allant de la cohérence, de la connexité, à la disparité et la dissemblance. Il en ira de même du travail formel évoluant dans le même espace. La forme pourra se construire, pourra émerger des divers chemins et sauts effectués pendant de nombreuses explorations. Ou alors celle-ci, déjà définie au préalable, viendra s'incarner dans un espace singulier qui aura alors donné chair à cette forme préfigurée.

## Mort et transfiguration

L'espace n'est pas inerte, il change au cours du temps. Parfois des paramètres deviennent inadéquats, ils ne font plus sens et parfois il apparaît que « quelque chose manque ». Il s'agit d'un espace ouvert, en aucun cas d'un espace clos. Il n'est pas question d'une simple accumulation d'éléments amenés là au gré du hasard. C'est un monde en devenir qui s'incrémente, en plus ou en moins, qui prend forme en même temps que le *projet musical* se construit. Il s'agrandit, s'altère et se modifie selon les besoins sans détruire pour autant son essence profonde.

Parfois au milieu d'une partition, en plein travail sur une cellule, on finit par n'utiliser que quelques uns des paramètres de la cellule en question. À d'autres moments la partition correspond, « paramètres par paramètres », à la cellule travaillée. Parfois un « accident » arrive, la cellule utilisée se doit de s'altérer, l'idée est trop « hors-cadre ». Mais tout du moins il y a une *préoccupation latente* qui demeure.

On pourrait discuter de la pertinence des paramètres de symétries harmoniques et rythmiques dans les simplexes d'attraction comme image d'un plus ou moins grand statisme. L'intégralité des paramètres que nous avons présentés pourraient être remis en question. Mais là n'est pas tellement le problème puisque chaque paramètre utilisé pour définir un simplexe peut être redéfini. Ils ont par ailleurs changé tout au long de notre recherche. Ce qui demeure pourtant et qui nous paraît le plus fondamental dans cette recherche, c'est cette manière d'envisager le matériau musical et son rapport à la forme, non plus comme un générateur infini de matière mais comme espace de constitution d'une pensée dynamique et relationnelle, comme moyen de constituer *dialectiquement* un monde de *devenirs*.

---

68 *Ibid.*

## Applications pratiques : *Sortir du noir et Namenlosen*

« *L'espace est un doute : il me faut sans cesse le marquer, le désigner ; il n'est jamais à moi, il ne m'est jamais donné, il faut que j'en fasse la conquête.*<sup>69</sup> »

L'un des projets principaux de cette thèse a été la composition de la pièce *Namenlosen*, créée en juin 2017 à la Cité de la Musique, Philharmonie de Paris et interprétée par l'Ensemble Intercontemporain (direction : Matthias Pintscher). Cette pièce pour 4 solistes (flûte, hautbois, trompette et alto), grand ensemble et électronique a été composée à partir de 4 espaces topologiques qui sont « parcourus » dans différentes parties. Chaque espace implique un temps particulier, un rapport particulier entre solistes et ensemble, des mises en lumière (au sens métaphorique) différentes selon le rôle défini pour l'électronique.

En voici la note de programme :

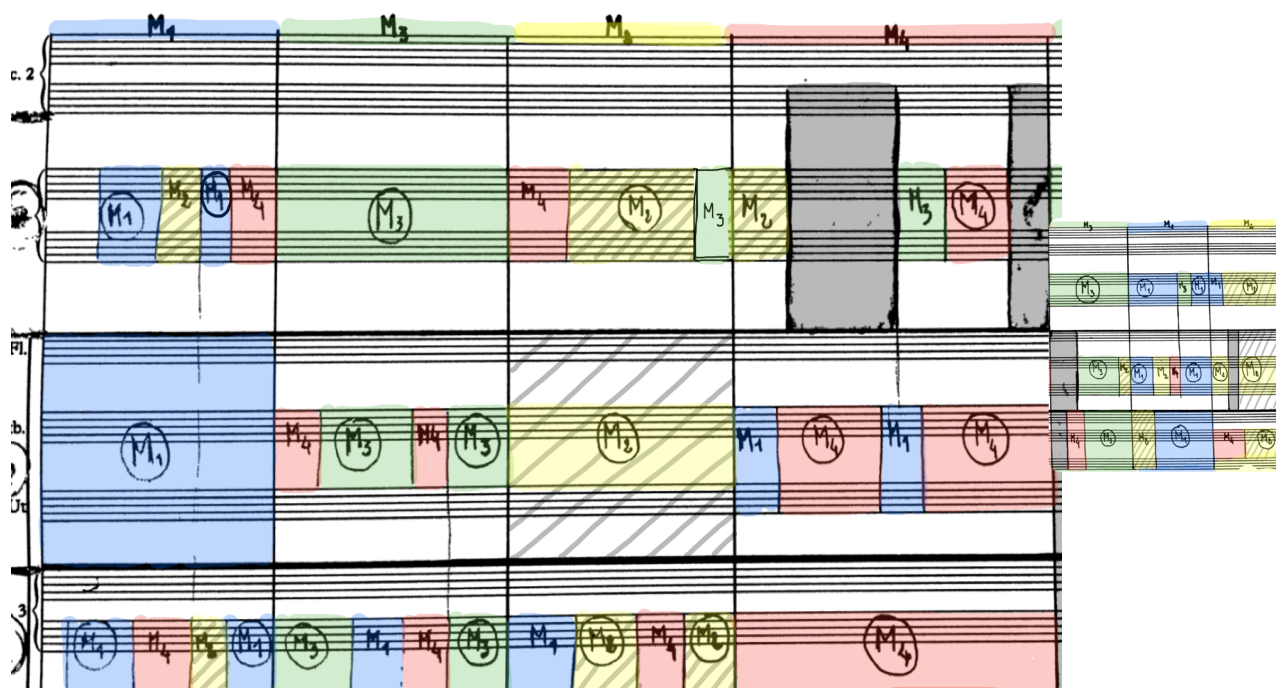
*Cette pièce est une tentative de renversement du mode de représentation de la figure du soliste. Disposés tout autour du public et de l'ensemble, les quatre solistes — les figurants devrait-on dire — coexistent avec une partie électronique qui forme avec eux une « force diagonale » faite tour à tour de signaux, d'appels, de (lignes de) fuites, de poursuites et de soulèvements. Confrontés à un ensemble parfois imposant, il s'agira pour eux de voir de quelle manière organiser leur coexistence, leur communauté et leurs survivances.*

*L'électronique joue à la fois le rôle de faisceau lumineux, de double et de contrepoint souple et réactif, capable de suivre et d'anticiper le jeu des quatre solistes et de faire lien entre leurs différentes temporalités. Parfois doublure des instruments (électronique d'ensemble), produisant un « être-dans un dehors » (« Le seuil, en ce sens, n'est pas autre chose que la limite, c'est pour ainsi dire l'expérience de la limite même, de l'être dans un dehors », G. Agamben, *La communauté qui vient*), qui se situe au-delà d'une simple addition du son instrumental et du son électronique. Parfois voix réelle (électronique soliste), contrepoint aux solistes, dans un temps commun ou différent, mais jouant du phrasé et de la souplesse que peut produire aujourd'hui une électronique réellement écrite. Parfois arrière-fond dans lequel l'ensemble se déploie et où les figurants viennent porter plainte au devant (« ne te plains pas, porte plainte ! » disait Heiner Muller). Si l'électronique fait donc ici éminemment partie du discours musical, elle s'avère être un médium radical de mise en scène.*

On peut voir dans le tableau suivant l'agencement des quatre espaces (M1, M2, M3 et M4) selon les parties<sup>70</sup>. Ces agencements se sont modifiés, « adaptés » de façon plus ou moins importante avec le temps, durant la composition, suivant les nécessités contingentes à l'écriture en train de se faire. Une partie contient un espace principal ainsi que des inserts d'espaces secondaires qui viennent former des sortes d'images du passé ou du futur selon le moment où l'espace principal lui correspondant a été ou lui sera présenté. Notons que solistes (2ème ligne), ensemble (1ère ligne) et électronique (3ème ligne) peuvent évoluer dans des endroits différents de l'espace, parfois même dans différents espaces dans le cas des espaces secondaires d'une partie. Ainsi la première partie se situe dans l'espace M1 mais contient des inserts des espaces M4 et M2.

<sup>69</sup> Georges Perec, *Espèces d'espaces*. Éditions Galilée, 2000.

<sup>70</sup> La représentation topologique des 4 espaces est donnée en annexe. Ainsi que les espaces qui ont servi aux différentes pièces composées pendant ce doctorat (*Tesla*, *Théâtre d'ombres*, *Sortir du noir*, *Namenlosen*).



Le plan présenté en page suivante est celui de la deuxième partie de *Namenlosen*, correspondant à l'espace M3 (en vert en deuxième colonne du tableau ci-dessus). Chaque case du plan reprend les éléments d'une cellule de l'espace auquel il fait référence. Le déroulement temporel n'est ici pas forcément proportionnel et a été défini et réévalué durant l'écriture. Les choix compositionnels se sont fait en « voyageant » d'un point de l'espace à un autre, en imaginant différents sauts ou en partant d'une idée musicale assez précise, en cherchant une cellule qui pouvait lui correspondre et en déduisant des développements à partir des cellules voisines. Mais ils peuvent aussi venir d'un choix purement arbitraire, d'une « envie compositionnelle » qu'on cherchera ensuite à situer dans l'espace. On peut distinguer des sortes de coordonnées<sup>71</sup> (ex : a308-t310-f308-g310) qui déterminent quelle cellule est utilisée en nommant ses 4 simplexes par une lettre déterminant une des 4 dimensions puis un nombre symbolisant une variable donnant accès aux 4 constituants (qui, en pratique dans l'application, se « révèlent » lorsqu'on clique dessus). Ces coordonnées sont celles entrées au préalable dans Mathematica pour construire les espaces compositionnels avec lesquels nous avons travaillé.

Ce modèle de plan permet de « fixer » les divers trajets dans un espace et d'organiser les différentes couches, à la fois entre elles mais également dans le temps. Il s'agit de voir les différents points de convergence (comme ceux qui sont entourés d'un liseré orange), les inserts d'autres espaces quand il y en a et d'avoir une *vision* plus lointaine avec une sorte de positionnement *devant le temps*. Tout cela est fait sur papier, mais l'on pourrait tout à fait imaginer une interface de travail liée à nos espaces et qui permette de placer des cellules dans le temps en affichant les variations de paramètres suivant le passage d'une cellule à une autre via des couches graphiques différentes.

<sup>71</sup> Notons que les lettres faisant référence aux simplexes correspondants diffèrent de celles que nous avons montré auparavant. Il s'agit d'un simple changement de dénomination des simplexes (« a » pour articulation est par la suite devenue « t » pour timbrique, « t » pour temporel est devenue « r » pour rythmique et « g » pour gravitationnel est devenu « a » pour attraction). Les dénominations que nous avons présenté jusqu'ici correspondant mieux, selon nous, au sens musical auxquelles elles réfèrent.



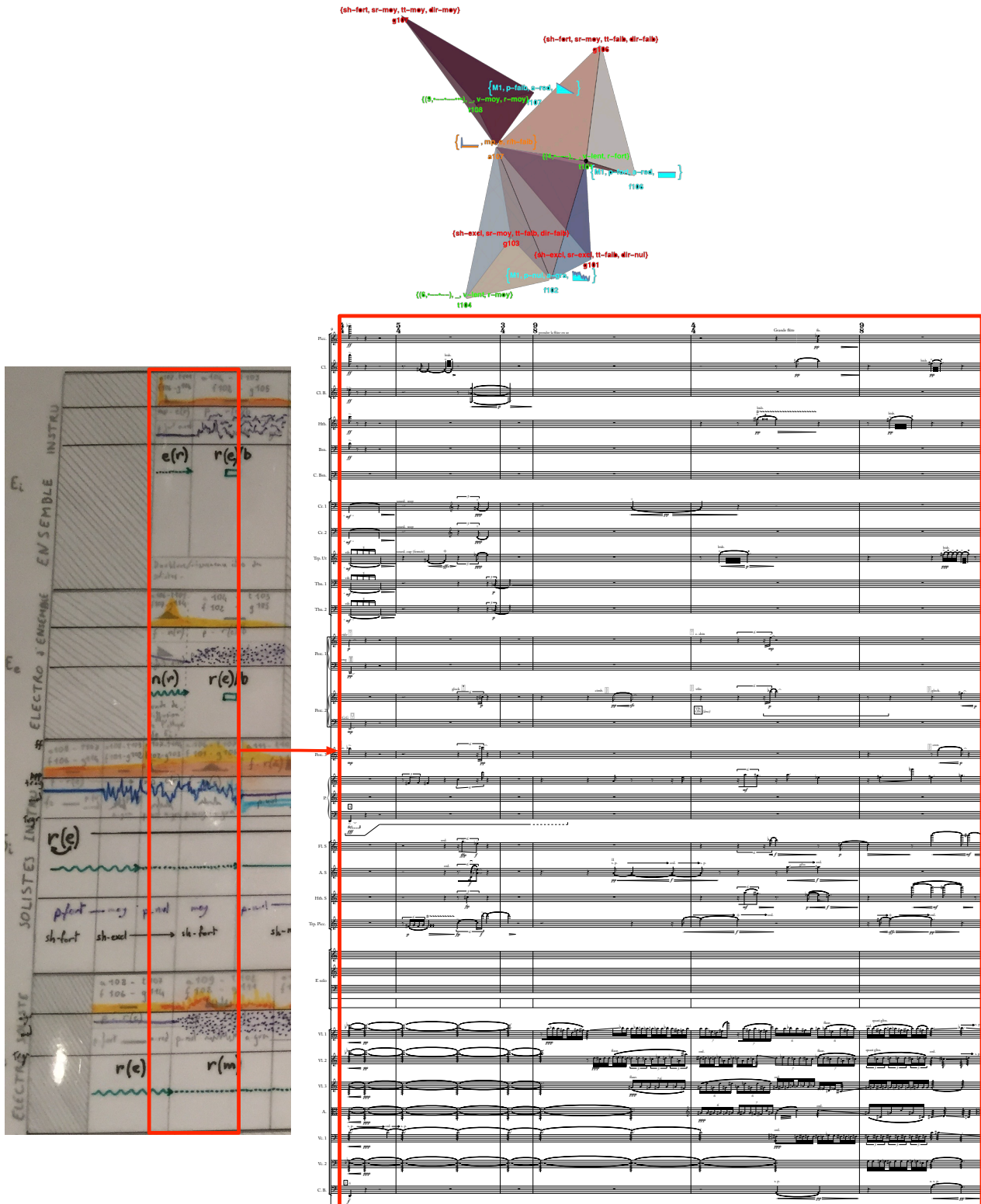
<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>	<p>4304-1304 f - n(e)</p>
<p>Electro Ensemble Instru</p>															
<p>Electro D'ensemble</p>															
<p>Electro Instru</p>															
<p>Electro Soliste</p>															
<p>Electro Soliste</p>															
<p>Electro de Salle</p>															
<p>Electro Soliste</p>															

Plan de la partie II de *Namenlosen* faisant référence à l'espace topologique lui correspondant (une cellule = une case du tableau).

## ORGANISER LE MULTIPLE : ESPACES COMPOSITIONNELS

Cette interface pourrait également servir « d'engin de navigation » dans nos espaces, notamment pour trouver des chemins dans l'espace satisfaisant certaines contraintes. Nous n'avons malheureusement pas pu développer un tel outil durant cette thèse et avons choisi de remettre cela à un travail ultérieur.

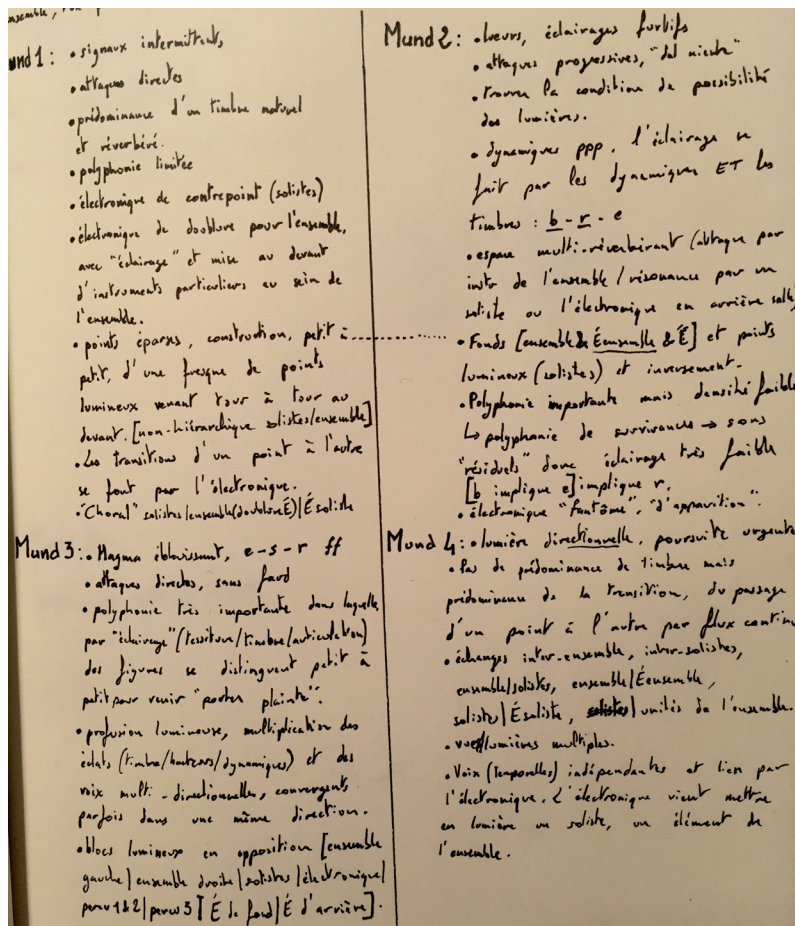
L'illustration ci-dessous montre l'interprétation faite dans la partition de cellules indiquées dans le plan de travail et qui correspondent notamment à deux cellules ayant un simplexe (timbrique) en commun. La structure topologique du dessus est un filtrage de l'espace M1 qui n'affiche que les cellules ayant pour élément commun le simplexe a107, qui caractérise notamment l'attaque/résonance du début jouée par l'ensemble.



La partie électronique, qui évolue dans ses cellules propres, n'est pas représentée sur la partition (uniquement sur le manuscrit pour des raisons indépendantes de notre volonté).

Nous avons eu pour parti pris de construire un espace pour chaque partie (Mund1 à Mund4), avec pour chacun d'eux des idées déjà bien établies quant aux rapports entre solistes, électronique et ensemble, ou au type « d'éclairage » des voix utilisés (voir la reproduction de notes ci-contre). La constitution de chaque espace s'est faite avec toutes ces spécificités en tête.

3 types différents d'électronique sont présents dans la pièce. L'électronique soliste est liée aux solistes et forme une voix supplémentaire, parfois hybride, capable de lier les solistes instrumentaux éloignés dans l'espace et de constituer un espace commun dans lequel les polyphonies s'installent. L'électronique d'ensemble n'est diffusée que dans les haut-parleurs de façade et se mêle à l'ensemble sur scène. Il s'agit d'une électronique « d'orchestration » doublant certaines parties instrumentales et profitant ainsi du mélange entre sons complexes instrumentaux et sons purs électroniques. L'électronique « de salle » ne joue que sur le fond. Présente dans toute la salle de concert, elle a pour rôle de créer des espaces dans lesquels l'ensemble, les solistes et l'électronique soliste peuvent coexister. Elle vient poser un arrière-fond permettant d'organiser un jeu de distance entre les différents types d'électronique. Ces différents types d'électronique, selon leurs spécificités, permettent de mettre en lumière chaque « personnage », qu'il soit soliste ou interprète de l'ensemble, pouvant passer du rôle de protagoniste à celui de figurant, quelque soit leur fonction de départ.



Les différents plans des parties de la pièce ont été réalisés grâce aux espaces correspondants qui ont été en conséquence le « lieu de travail » et d'édification formelle. Certaines cellules ont été ajoutées durant la composition de la pièce et le passage à l'écriture a parfois amené à une reconfiguration du plan d'origine, notamment dans les superpositions entre ensemble, solistes et électronique. Il s'agissait d'ajuster la densité et la complexité des différentes couches pour éviter certaines surcharges. Les espaces compositionnels permettent entre autres choses d'assurer une sorte de cohérence dans le discours musical et de voir, en passant d'une cellule à l'autre, les éléments qui sont conservés et ceux qui ne le sont pas. Les cellules, si elles sont interprétables, permettent cependant, grâce à l'ensemble des paramètres qui les constituent, d'assurer une identité musicale forte et le médium de réflexion que constitue l'espace compositionnel permet de penser en permanence au contexte et aux différents agencements possibles.

## Un exemple d'interprétation de cellules dans *Sortir du noir*

La pièce *Sortir du noir*, pour violoncelle et électronique, écrite en 2016 pour Séverine Ballon, a été écrite à partir des espaces M2 et M3 de *Namenlosen*. Nous présentons ici deux esquisses réalisées à partir de deux cellules différentes. La première se concentre sur la cellule [a311-t301-f306-g305] dont les paramètres sont les suivants :

- simplexe fréquentiel : matrice d'intervalle M3<sup>72</sup>, polarisation faible, ambitus moyen (non respecté, remplacé par un grand ambitus), profil descendant.
- simplexe rythmique : métrique de phrase [ - - ●●●● - ] ici divisée en 4 mesures [ -|●●●●| - ], tempo fixe, vitesse moyenne, régularité forte (peu respecté).
- simplexe timbrique : enveloppe « bloc », dynamique moyenne *ff*, timbre principal saturé, timbre secondaire enrichi (ici travail sur les harmoniques et multiphoniques du violoncelle) et degré d'hétérogénéité fort.
- simplexe d'attractions : symétrie harmonique et rythmique faible/nulle, tension timbrique forte et directivité moyenne.

The image shows a complex diagram at the top and a musical score below it. The diagram is a 3D-like structure with several vertices and edges, each labeled with musical parameters in various colors. The labels include:
 

- Top-left: ((S, ←, ←, ←, ←), V, -traj, r-moy)
- Top-right: (M3, n-falk, a-moy, )
- Middle-left: (sh-falk, sr-nul, tt-fest, sh-moy)
- Middle-right: (sh, sr-nul, tt-fest, sh-moy)
- Bottom-left: (sh-moy, sr-nul, tt-fest, dir-fort)
- Bottom-right: ((S, ←, ←, ←, ←), V, -traj, r-moy)
- Bottom: (M, n-falk, a-moy)

 The musical score below is for a cello (Vcllo) and features a complex rhythmic pattern with many notes. Above the score, the cell identifier [a311-t301-f306-g305] is written. The score includes dynamic markings like *ff* and *fff*, and various articulation marks.

On voit dans cet exemple que la cellule est appliquée à une phrase relativement longue. La métrique n'est pas utilisée comme un rythme mais comme un mode d'organisation de la phrase. Le type d'interprétation d'une cellule dépend donc fortement de l'échelle à laquelle elle est appliquée. Il n'est pas rare qu'un ou plusieurs des paramètres se trouvent modifiés, comme ici l'ambitus qui s'est vu élargi. Cela n'altère cependant que très peu l'identité de la cellule de départ.

72 Matrices et espaces compositionnels sont disponibles en annexes, avec les programmes correspondants.

L'esquisse suivante utilise la cellule [a306-t309-f310-g308], correspondant aux paramètres suivants :

- simplexe fréquentiel : matrice d'intervalle M3, polarisation nulle, ambitus réduit, profil mixte.
- simplexe rythmique : métrique de phrase [ • – – • – – •• – ], tempo fixe, vitesse rapide et faible régularité.
- simplexe timbrique : enveloppe attaque-decrescendo progressif, dynamique moyenne *mp*, timbre principal naturel, timbre secondaire enrichi (ici *sul pont.*) et degré d'hétérogénéité moyen.
- simplexe d'attractions : symétrie harmonique forte, symétrie rythmique faible, tension timbrique faible et directivité moyenne.

Ici c'est la polarisation qui a été altérée puisqu'elle n'est pas « nulle ». Ce choix s'est fait de façon à lier sémantiquement cette cellule à une suivante qui représentait un saut de l'espace M3 à l'espace M2 — lui caractérisé par des polarisations relativement importantes. L'interprétabilité des cellules nous permet ici de tenir compte en permanence du contexte dans lequel elles s'inscrivent et de prendre en conséquence les décisions qui s'imposent.

La page suivante est la reproduction d'une « feuille de travail » dans laquelle il s'agissait de travailler sur une suite de cellules — en préparation de *Sortir du noir* — en faisant éventuellement plusieurs versions d'une même cellule. Chaque cellule a été retravaillée puis *remontée* par la suite, de manière à trouver le meilleur agencement possible.

ORGANISER LE MULTIPLE : ESPACES COMPOSITIONNELS

The image displays a series of musical score pages and 3D geometric diagrams. The score pages are arranged in a grid-like fashion, with some pages featuring handwritten annotations and dynamic markings such as *pp*, *p*, *f*, and *ff*. The 3D diagrams, located at the top and bottom of the page, use colored planes (purple, blue, red, yellow) to represent spatial relationships between musical elements. A handwritten "X15/2" is visible in the middle section. The overall layout suggests a process of organizing multiple compositional spaces.

## Quelques prolongements extra-musicaux...

« Quand je lis un livre sur la physique d'Einstein auquel je ne comprends rien, ça ne fait rien : ça me fera comprendre autre chose ».

Pablo Picasso

Les relations spatiales ont été utilisées, dans le passé et jusqu'à aujourd'hui, pour des utilisations différentes, qui permettent notamment d'organiser et de structurer la représentation par *analogie* avec l'objet représenté. Rudolf Arnheim soutient<sup>73</sup> que les opérations cognitives viennent directement de notre perception du monde, la vision servant de système sensoriel paradigmatique.

Nous donnons, à titre d'exemples, quelques opérations et représentations de ce type :

**Carte symbolique.** Une carte d'état major ressemble à l'objet qu'elle représente à cause de ses caractéristiques métriques. Mais une carte des stations du métro parisien est une représentation diagrammatique. En effet, établir un chemin entre deux stations est une tâche purement topologique qui ne s'intéresse pas à la métrique mais seulement à la connexité des segments de ligne empruntés.

Un autre exemple de carte symbolique est donné par la « carte de Tendre » où la géographie d'un pays imaginaire reflète métaphoriquement l'organisation des concepts d'un domaine donné (ici, l'amour).

**Les représentations graphiques d'objets logico mathématiques.** Plusieurs objets ou concepts mathématiques admettent une représentation spatiale ou géométrique :

- diagrammes de Venn pour les formules logiques
- diagramme sagittal pour la représentation des relations
- graphe de Cayley pour les groupes
- représentation graphique d'une fonction réelle (cette représentation est diagrammatique en ce sens que l'intérêt porte sur l'allure de la courbe, points asymptotiques, limites, points singuliers, croissance, décroissance plutôt qu'à des valeurs exactes)
- les diagrammes de la théorie des catégories, une opération classique consistant à compléter le diagramme (preuve de l'existence d'un morphisme)
- représentation graphique des arbres (un arbre est un terme d'une algèbre)
- diagramme de Hasse pour la représentation d'un treillis

Remarquons que les objets de cette liste n'ont pas une nature intrinsèquement spatiale. Mais nous pouvons *a fortiori*, ajouter tous les objets développés pour la formalisation de l'espace, savoir, toute la topologie et la géométrie.

**Diagrammes d'état en physique.** Les diagrammes d'état en physique permettent de représenter spatialement le comportement de la matière. Par exemple, dans les diagrammes de changement de phase, les régions traversées par une courbe, indiquent les états successifs de la matière. Dans les diagrammes de Feynman, ce sont les interactions entre particules élémentaires qui sont représentées de manière spatiale.

**Schémas fonctionnels.** Les circuits électriques, les graphes dataflow, les graphes de Bond sont la représentation par des graphes de systèmes formés d'éléments fonctionnels.

<sup>73</sup> Rudolf Arnheim, *Visual Thinking*. University of California Press, 1969. Trad française : « La pensée visuelle », Champs-Flammarion, 1976.

**Méthodes d'analyse.** Plusieurs méthodes d'analyse et de conception de systèmes, tant matériel que logiciel ou organisationnel, font appel à des graphiques pour représenter des flux d'information et la structure du système, plusieurs échelles et selon des paramètres différents.

**Les jeux.** Les jeux de plateau comme les Dames, les échecs, le Go, sont décrits par des configurations spatiales de pièces et l'arbre du jeu permet de rendre compte de la situation des coups.

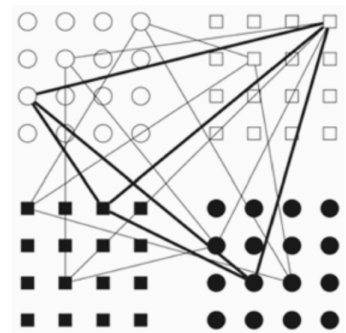
**Mémoire associative.** Plus récemment, suite à une séance de travail au Collège de France<sup>74</sup> et à une présentation de notre modèle, nous avons pu constater un certain nombre de similitudes avec la démarche de Claude Berrou<sup>75</sup> qui consiste, entre autres choses, à modéliser le réseau cortical et la mémoire associative.

*« Du côté des réseaux de neurones récurrents, une innovation majeure, au moins conceptuellement, fut proposée en 1982 par John Hopfield pour viser la fonctionnalité de mémoire associative. [...]*

*Pour expliquer la mémoire cérébrale, le principe auquel on se réfère le plus souvent est donc celui de la mémoire associative qui est un dispositif dans lequel il est possible de retrouver les messages qu'on y a stockés à partir de fractions de leurs contenus, même partiellement approximatives ou erronées. Par opposition avec les mémoires indexées qui assignent une place spécifique à chaque message, et dont l'adresse est le seul lien pour le retrouver, les mémoires associatives utilisent des structures qui mutualisent le matériel disponible. Les messages y sont donc physiquement chevauchants, ce qui offre des possibilités d'association et de croisement d'informations, susceptibles d'ouvrir vers des aptitudes à l'analogie et à l'élaboration. »<sup>76</sup>*

Dans son modèle, un « morceau d'information », dit *infon*, est formé par le couplage d'un certain nombre de caractères quantifiés. Il peut être comparé à ce que Jean-Pierre Changeux, l'un des premiers défenseurs de la théorie de l'activité neurale par populations, appelle un percept primaire<sup>77</sup> ou encore au métapheur de Julian Jaynes, objet mental concret sur lequel la raison construit ses métaphores d'ordre supérieur<sup>78</sup>.

*« Pour une présentation commode des réseaux de grappes et de cliques, supposons que toutes les cliques ont le même ordre, c'est-à-dire le même nombre de sommets  $c$ , et que l'intensité de chaque caractère peut être évaluée sur  $l$  niveaux. Dans le cortex cérébral, la valeur de  $l$  est très variable selon la précision requise dans la mesure du caractère, sans doute nettement plus élevée pour quantifier la luminosité d'une couleur que pour attribuer une intensité au goût du salé. Mais ici, nous faisons simple : toutes les grappes ont le même cardinal  $l$ . Le nombre total de nœuds du graphe multi-parti est donc  $n = c \times l$ . La figure décrit un tel dispositif pour  $c = 4$  et  $l = 16$ . Chaque grappe de nœuds (cercles pleins ou vides et carrés pleins ou vides) représente un caractère, un éventail de mesures ou attributs exclusifs les uns des autres. Pour réaliser l'apprentissage d'un infon, les attributs de tous les caractères sont identifiés par autant de nœuds du graphe – un par grappe – puis reliés les uns aux autres pour former une clique spécifique. Puisqu'il ne peut y avoir qu'un seul attribut dans chaque classe,*



<sup>74</sup> Groupe de travail sur la notation organisé par Gérard Berry et Philippe Manoury.

<sup>75</sup> Claude Berrou est un chercheur français en électronique et informatique, surtout connu pour les turbo codes utilisés notamment en téléphonie mobile. Son activité de recherche est maintenant presque entièrement consacrée aux neurosciences computationnelles.

<sup>76</sup> Claude Berrou & Vincent Gripon. *Petite mathématique du cerveau*. Éditions Odile Jacob, 2012.

<sup>77</sup> Jean-Pierre Changeux, *L'Homme neuronal*. Fayard, 1983.

<sup>78</sup> Julian Jaynes, *La Naissance de la conscience dans l'effondrement de l'esprit*. PUF, 1994



*les cliques ont toutes le même ordre  $c$ , soit quatre dans l'exemple de la figure. »<sup>79</sup>*

Les cliques dont parle l'auteur peuvent être associées dans notre cas à un simplexe ou éventuellement un complexe simplicial. Elles ont toutes le même ordre, de la même manière que nos simplexes sont tous de dimension 3.

---

<sup>79</sup> Claude Berrou, *opus cit.*

# LA DIALECTIQUE DES TEMPS

« On pourrait se demander si la rationalité intégrale à laquelle tend la musique est simplement compatible avec la dimension du temps ; si ce pouvoir de l'équivalent et du quantitatif que représente la rationalité ne nie pas au fond le non-équivalent et le qualitatif dont la dimension du temps est séparable.<sup>80</sup> »

Le chapitre qui suit est le fruit de réflexions menées sur la question du temps en composition. Celui-ci sera mis en perspective des précédents chapitres, notamment celui qui concerne les espaces compositionnels. Il ne traite en aucun cas du temps en général et n'a pas la prétention d'une quelconque portée philosophique. Les articles et ouvrages recensant les diverses conceptions du temps dans l'histoire de la philosophie sont suffisamment nombreux pour que nous puissions nous passer ici de cet exercice et nous concentrer sur les aspects musicaux et compositionnels qui sont l'objet de cette thèse.

Si les questions du temps, du matériau, de la forme et de leurs relations sont des questions centrales — et depuis bien longtemps — pour les compositeurs, il importe de se demander si la vieille dichotomie entre forme et matériau n'est pas à repenser en regard des nouveaux dispositifs susceptibles d'amener une appréhension à la fois plus fine et plus globale des diverses dimensions temporelles. Dans un tel contexte, et dans le cas plus précis de la musique mixte, la question que nous posons ici est la suivante : dans quelle mesure l'existence d'un continuum entre les notions de forme et de matériau est-elle liée à la composition consciente d'entrelacs temporels ? Nous entendons par là que la conscience du temps musical, si essentielle au compositeur, réside pour nous dans une volonté préméditée de façonner un tissu de relations temporelles au sein duquel s'inscrit, par-delà les concepts de forme et de matériau, une trajectoire protensionnelle : une ligne sous-jacente aux entrelacs, intentionnalité anticipant les événements à partir de ce qui est, et de ce qui a été.

À travers le concept d'entrelacs temporels, que nous développerons tout au long de ce chapitre, nous tenterons d'esquisser une dialectique des temps en composition, indispensable selon nous à l'élaboration d'une pensée musicale aujourd'hui. Si la question du temps musical paraît souvent insaisissable et si une quelconque définition ne peut, au bout du compte, n'être que provisoire, il nous semble urgent, pour paraphraser Beckett, « *d'essayer dire* ».

À titre d'introduction, nous traiterons tout d'abord la question des différents types de temps auxquels le compositeur est confronté dans son travail. Qu'il soit lié à l'usage d'outils, de techniques particulières, ou tout simplement à la pratique sur le papier, le travail compositionnel implique souvent des déformations dans la perception temporelle. Nous en étudierons quelques unes. En nous dirigeant ensuite vers la composition en elle-même et non sa pratique, nous tenterons de définir ce que nous appelons une *qualité temporelle*. Nous verrons que la construction d'un temps particulier relève de la constitution d'un complexe de temporalités synchronisées entre elles, qui n'est pas seulement défini par des données rythmiques ou de tempo mais qui dépend d'un agencement local comportant des évolutions qui lui sont propres. Nous verrons que la superposition de plusieurs couches temporelles et la composition de leurs synchronicités tend à supplanter, en terme de singularité, la définition des couches en elles-mêmes. Tout au long du chapitre, il sera intéressant de noter, en ayant en tête nos propos sur les espaces compositionnels, les nécessités de changements d'échelles réguliers.

<sup>80</sup> Theodor W. Adorno, *Quasi una fantasia*.

## Multiplicités des temps en composition

La question de la multiplicité des temps se retrouve à divers moments de la composition. Nous distinguerons les temps inhérents au processus compositionnel et ceux mis-en-œuvre au sein même de la composition, ceux du « temps composé ». Les temporalités liées à l'interprétation seront seulement évoquées, le présent chapitre ayant pour but de se concentrer sur les questions d'ordre compositionnel.

Le processus compositionnel impose, peut-être davantage aujourd'hui, des étapes de travail qui impliquent de multiples rapports au temps. Qu'il s'agisse du temps « spéculatif » — qui se déploie durant toute la composition d'une pièce dans le cas d'une œuvre écrite à l'origine sur papier — ou des temps liés à un outil particulier d'aide à la composition ou, dans le cas d'une musique électroacoustique ou mixte, à un outil de génération de son ou de montage, le compositeur est confronté à une multiplicité de types temporels différents. Du temps « subi » au temps « composé », la palette de nuances s'avère gigantesque.

Les différents outils mis à disposition des compositeurs depuis l'apparition de l'informatique — synthétiseurs, séquenceurs et bancs de montage, logiciels d'aide à la composition et autres outils d'analyse et de traitement du signal — amènent avec eux des rapports au temps particuliers impliquant un certain nombre de conséquences à prendre en compte dans la composition.

### Hors-temps, aide à la composition et mise-en-temps

Bien des outils d'aide à la composition représentent pour certains compositeurs un terrain d'expérimentation, autant du point de vue du matériau en général que du point de vue structurel. Ces outils intègrent le plus souvent des paradigmes atemporels et proposent des maquettes ou simulations temporelles de « plans compositionnels ». Une première phase de préparation du matériau se fait alors de manière atemporelle, jouant des puissances combinatoires et algorithmiques de ces outils devenus « crayons augmentés » pour générer un matériau plus ou moins personnalisé. Il est évident que des questions temporelles peuvent entrer en jeu dans cette première phase, mais nous dirons qu'elles se posent souvent a posteriori, dans une phase de « mise-en-temps », souvent associée directement à l'écriture sur partition. Du schéma de départ éclot un choix de temporalisation.

La question que l'on est en droit de se poser est de savoir dans quelle mesure les outils d'aide à la composition développés depuis les années 1990 sont en mesure de favoriser une meilleure appréhension des problématiques temporelles durant le travail de composition. Développés et pensés dans une période où la question du matériau était prédominante, ils paraissent souvent limités aujourd'hui dans leur faculté à « parler du temps ».

Les outils d'aide à la composition ont souvent pris le parti de la représentation graphique pour faciliter leur accès à des compositeurs qui n'étaient pas forcément familiers avec la technologie. C'était aussi un moyen ergonomique d'organisation qui, aujourd'hui encore, paraît tout à fait utile aux compositeurs. Le problème de ces représentations graphiques demeure cependant dans les limitations de leurs représentations qui peuvent devenir des limitations de compositions. Si l'on souhaite composer des processus complexes, évoluant de manière polyphonique mais étant synchrones sur seulement quelques éléments, et si l'on imagine que ces processus sont dynamiques et peuvent évoluer de manière récursive, on se rend rapidement compte que leur représentation graphique devient pratiquement impossible (à moins de trouver de nouveaux types de représentation, problématique fondamentale aujourd'hui). Si l'on passe à des langages informatiques non graphiques, ces limitations semblent beaucoup moins grandes mais alors la composition de processus complexes devient compliquée à réaliser lorsque justement la représentation graphique n'est plus possible. Peut-on alors composer des temps — mettant en jeu des complexes d'évolutions dont les composantes sont plus ou moins interdépendantes — sans en avoir une représentation visuelle ? La représentation devient alors à la fois mentale, textuelle et, d'une certaine manière expériment-

tale, nécessitant une confrontation à un « résultat », tout éphémère soit-il. La question demeure toujours la même : comment *organiser* le temps, comment en avoir une certaine représentation, ne serait-ce que mentale, de façon à aborder correctement les questions inhérentes à la forme.

Le travail des compositeurs spectraux, utilisateurs fréquents des outils d'aide à la composition, lorsqu'il intégrait le temps comme composante du matériau (dilatation, compression temporelle...) s'est parfois limité à réaliser un certain type d'interpolation entre un état A et un état B dans une durée donnée. Il n'était pas pour autant évident qu'il y ait une réelle définition temporelle dans la composition. La complexité résidait davantage dans la définition du timbre que dans celle du temps. Il ne suffit pas d'intégrer des fonctions générant des modulations rythmiques pour pouvoir parler de temps musical. Aujourd'hui encore, la question du matériau, prédominante (aide à la composition, aide à l'orchestration), semble éclipser celle du temps. Il est possible que ce biais soit lié aux méthodes de représentation, restées relativement classiques. Si « les limites de mon langage signifient les limites de mon propre monde », il faut se demander si la relative uniformité des représentations temporelles n'est pas à mettre en cause. Nous discuterons de cette question plus précisément dans la partie dédiée à l'organisation des temps en musique mixte, notamment par l'utilisation du langage Antescofo.

## Travail du son, timeline et chronomètre...

Dans le cas du son en général, nous avons beau travailler avec un matériau qui, intrinsèquement, se déroule dans le temps, nous sommes toujours confrontés à des échelles de temps très hétérogènes, parfois même contradictoires. Ainsi, un son s'inscrivant dans une certaine échelle de temps pourra se retrouver dans une échelle tout à fait différente à partir du moment où il sera superposé à un autre son qui par sa présence altérera la perception temporelle du premier<sup>81</sup>. Et pourtant dans la situation assez classique du banc de montage — qui n'a comme modèle de temps que celui de la séquence — le compositeur se retrouve devant une symbolisation figée et linéaire : « la timeline », le temps scientifiquement chronométré.

On pourrait penser que celle-ci résout le problème et que le compositeur n'a plus qu'à superposer ses sons, au gré de ses envies, sans se soucier de savoir comment le matériau se déploie dans le temps puisqu'à priori, montant son matériau, le compositeur l'inscrit dans le temps et en a une perception a priori objective. Mais la timeline ne garantit en aucun cas une meilleure gestion des temporalités. En cela, la profusion de compteurs chronométriques qu'on trouve dans les logiciels montre sa pleine vacuité. Ces compteurs ne peuvent s'utiliser que de façon très technique (point de montage par exemple) et ne donnent aucune perception ou même information sur un quelconque déroulement temporel. La timeline n'est donc ni plus ni moins qu'un outil générique qui, certes, nous donne un aperçu du réel, mais en aucun cas ne nous apporte une solution quant à la question du *temps composé*.

Il y a par ailleurs dans ces séquenceurs un aspect qui peut fausser le rapport au temps. Il s'agit du moyen par lequel nous pouvons « entrer dans une parcelle de temps » souvent très restreinte : le zoom. Si la timeline semble nous donner un temps unifié, la possibilité de zoomer et donc de travailler sur des blocs de temps particuliers met largement à mal cette affirmation. Le temps devient élastique, car le zoom n'est pas uniquement un gadget graphique. Il implique, par le fait que nous pouvons grâce à lui nous concentrer sur une parcelle de temps parfois microscopique, d'étirer pendant un certain moment du travail l'attention auditive sur un très bref passage. Nous en avons tous fait l'expérience en travaillant pendant un certain temps sur une partie zoomée, sur une portion de temps. Nous agençons nos sons tel que nous l'entendons, écoutons le résultat dans cette échelle de temps et cela nous satisfait. Pourtant, il arrive qu'en repassant sur une échelle d'écoute plus grande, le montage « ne marche plus ». C'est qu'en étirant le temps pour y percevoir ses plus infimes fluctuations, nous dé-contextualisons les forces en présence. Et

81 La problématique est la même pour la composition instrumentale, et notamment l'orchestration.

comme avec l'élastique, quand nous tirons fortement dessus et que nous relâchons prise, il y a comme une onde qui se propage en s'amortissant sur le support et non pas seulement sur la partie étirée. Quand nous travaillons sur une parcelle très réduite de temps, que nous agençons nos sons, nos enveloppes, dans ce temps « zoomé » particulier, nous changeons résolument la perception y compris temporelle des éléments qui l'entourent. Il y a interdépendance des temporalités.

Nous nous approchons ici d'un concept découvert au cinéma par Lev Koulechov en 1922. L'effet Koulechov désigne la propension d'une image à influencer sur le sens des images qui l'entourent dans un montage cinématographique. Ainsi, les images ne prennent sens que les unes par rapport aux autres, et le spectateur est amené inconsciemment à les interpréter dans leur succession et non dans leur indépendance. Cet effet est à la base de la narration cinématographique. Il y a donc *contamination sémantique*.

Ces phénomènes sont assez proches de celui dans lequel le compositeur, penché sur son manuscrit, se concentre sur une mesure particulière pendant un temps plus ou moins long. La perception temporelle de cette mesure, ainsi que les mesures qui l'entourent, est altérée par la focalisation réalisée sur celle-ci.

## Particularités de la musique de synthèse

Lorsqu'un instrumentiste joue une note tenue, notre inconscient nous prédit qu'il va finir par s'arrêter. Nous connaissons le « code » implicite de son temps. On sait par ailleurs que, selon les cultures, ce code tend à diverger. Nous citerons à titre d'exemple, la percussion du gagaku qui vient ponctuer les étirements de temps si caractéristiques à cette musique. A chaque entrée du shō, nous attendons le taïko pour nous tourner vers la grande périodicité suivante. Souffle de l'instrumentiste, longueur de l'archet, fin d'une résonance, nous connaissons a priori leur fin. Face au son de synthèse, la potentialité d'un infini est toujours sous-jacente (la musique dite « de drone » en sait quelque chose...), la perception du temps est donc radicalement différente de celle de l'écoute d'un instrument réel. D'une certaine façon, nous marchons avec la synthèse sur un terrain totalement vierge de stigmates temporels, mis à part peut être les sédiments de notre inconscient venant parasiter notre perception de la temporalité. Par association inconsciente, nous pouvons à certaines occasions faire des liens entre des entités pourtant distantes et par cela brouiller les repères temporels.

La musique de synthèse implique un rapport particulier au continu et donc des potentialités temporelles différentes de celles que l'on peut trouver en musique purement instrumentale ou dans des sons exclusivement concrets. Les possibilités de contrôle étant extrêmement grandes, l'écriture d'une électronique de synthèse amène à composer des évolutions de paramètres évoluant parfois chacun dans son propre temps et se situant souvent à des échelles temporelles différentes. Il apparaît alors, mais cela est valable pour la composition en général, que la construction d'un temps particulier relève de la constitution d'un complexe de temporalités synchronisées entre elles de manière à donner une identité temporelle particulière à cette construction. L'agencement des temporalités supplantant alors, dans une quête de singularité temporelle, la définition des couches en elles-mêmes.

Si les questions temporelles sont généralement les mêmes en composition instrumentale et en musique mixte, il demeure des spécificités — liées aux possibilités inhérentes à l'informatique — propres à la musique mixte. Le développement des langages synchrones apporte notamment de nouveaux moyens de « parler du temps ». Les possibilités de composition de processus dynamiques et réactifs, prenant en compte l'interprétation du musicien, mais aussi les possibilités de dialogues entre processus amènent à la composition une dimension temporelle renouvelée. Nous en discuterons dans la partie dédiée à l'écriture du temps en musique mixte.

## La notion de qualité de temps

Nous avons parlé succinctement des différents types de temps auxquels un compositeur est confronté dans le processus compositionnel. Alors même qu'il doit trouver des moyens pertinents musicalement pour agencer les temporalités qui sous-tenderont toute sa musique, le compositeur doit prêter attention aux différents biais de perception imposés de façon plus ou moins consciente par les outils qu'il utilise mais aussi par sa propre pratique. Il nous semblait important d'en montrer quelques exemples avant d'attaquer le cœur de notre propos, celui de l'organisation des temps dans une œuvre musicale. Il ne s'agira pas de faire une liste des différents types de temps, ni de présenter des techniques de superposition. Nous essaierons plutôt de voir comment une qualité de temps se définit et comment l'organisation d'un matériau est susceptible d'influer sur celle-ci. Nous verrons comment les diverses échelles de temps participent à la constitution des qualités de temps d'une pièce musicale.

On parle souvent de qualité de temps pour définir la façon singulière qu'a le temps de s'écouler à un moment donné. Notion floue et néanmoins fondamentale en composition, elle n'est souvent traitée que de manière très allusive. Il nous semble pourtant primordial de se demander de quelle manière une qualité de temps se définit et se crée — quand bien même, encore une fois, les réponses trouvées s'avèreraient toutes provisoires et incomplètes.

### Concordance des temps

*« Seul le présent existe dans le temps, et rassemble, résorbe le passé et le futur ; mais le passé et le futur seuls insistent dans le temps, et divisent à l'infini chaque présent. Non pas trois dimensions successives, mais deux lectures simultanées du temps. »<sup>82</sup>*

Nous avons parlé, tout au long de cette partie, des nécessités de passages, lors de la composition, à différentes échelles. Si l'on en revient à nos espaces, il y a l'échelle du simplexe pris dans sa singularité, il y a l'échelle du simplexe intégré à son espace, mis en regard de ses voisins, plus ou moins proches, et enfin il y a l'échelle qui agence tous les simplexes dans un temps bien plus grand, qui organise les éventuels sauts d'un espace à l'autre, qui manie les oppositions, les chemins et les différentes trajectoires qui traverse de façon plus ou moins marquée l'intégralité d'une pièce : celui de la forme. Nous avons vu combien l'organisation du matériau et ses liaisons à la constitution formelle pouvaient être intriquées.

Si nous voulons nous concentrer sur la question du temps, alors posons la question suivante : au sein même de la partition, quel élément, quel processus fait lien entre plusieurs temporalités et comment organiser à la fois leur structure interne et la structure de leurs synchronicités ? Il s'agit bien là d'un problème de contrepoint de structures, voir plus généralement d'un contrepoint de qualités temporelles. Des couches temporelles peuvent demeurer totalement indépendantes, elles n'en restent pas moins liées par le fait qu'elles sont superposées et que de ce fait, elles occupent un temps commun, d'échelle supérieure. C'est l'organisation des temporalités qui importe ici — qu'il s'agisse de celles inscrites sur le papier ou dans un programme — le point critique reste celui de la concordance des temps.

Il nous apparaît que l'organisation temporelle se divise en trois types, en trois échelles que seraient celles du micro — de l'organisation interne d'une voix — du méso — organisation des voix entre elles, de leurs synchronicités — et du macroscopique. Ces trois échelles de temps, nous le verrons tout au long du chapitre peuvent s'imbriquer, l'une étant la plupart du temps liée, intriquée aux autres. Il ne s'agit pourtant pas d'un simple jeu de poupée gigogne. Nous verrons qu'encore une fois ici, se met en place un tissu de *relations* et que ces différentes échelles sont faites, avant toute chose, d'entrelacements.

<sup>82</sup> Gilles Deleuze, *Logique du sens*. Paris, Éditions de Minuit, 1979. p14.

### Organisation interne à une voix

Le premier type d'organisation temporelle est celui qui détermine la structure temporelle interne d'une voix. Le terme « voix » est ici pensé comme détermination d'une unité musicale autonome participant seule ou avec d'autres au discours musical d'un passage donné ou autrement dit à la fabrication d'un instrument, d'une structure sonore. L'identité temporelle d'une voix n'est ici en aucun cas exclusivement déterminée par des notions rythmiques ou liées au tempo. Celle-ci réside dans une conjonction de paramètres évoluant dynamiquement. Nous tenons ici à rappeler au lecteur la raison qui nous avait amené à imaginer ce principe d'espaces compositionnels dont nous avons parlé. Il s'agissait justement de formaliser cet ensemble de relations qui font qu'un groupe de paramètres plus ou moins abstraits peut devenir une entité musicale à part entière. La définition de la qualité temporelle d'une voix implique donc la composition d'un réseau de relations entre des paramètres hétérogènes — paradigme de simultanéité — et de son évolution temporelle — paradigme de durée. Ce réseau de relations et son évolution détermine le premier type d'organisation temporelle : celui de la voix individuelle et singulière.

Les images ci-dessous présentent 2 exemples simples de « voix temporelle ». L'exemple de gauche est tiré de la pièce *Nachleben*<sup>83</sup> et se compose de 4 voix : une réalisée par la soprano et le baryton, une deuxième par le percussionniste et 2 autres par l'électronique. L'exemple de droite<sup>84</sup> montre une seule voix temporelle jouée par 4 percussionnistes différents.

The image displays two musical score examples. The left example, from *Nachleben*, features a complex arrangement of four 'temporal voices' in 5/4 time. It includes staves for soprano (S), baryton (B), and percussion (P), with dynamic markings like *mf*, *f*, and *ff*. The right example, from *Théâtre d'ombres*, shows a single 'temporal voice' in 3/4 time, distributed among four percussionists, with dynamic markings such as *p*, *pp*, *mp*, *ffz*, and *sfz*.

A gauche un agencement de 4 « voix temporelles » dans *Nachleben*, pour soprano, baryton, percussions et électronique. A droite, une seule voix temporelles répartie entre plusieurs interprètes (*Théâtre d'ombres*, pour 6 percussions).

On remarque ici que deux voix (ici instrumentales), prises cette fois dans le sens contrapuntique du terme, peuvent ne former qu'une seule voix temporelle dans la mesure où elles partagent un nombre important d'éléments communs qui les rendent peu dissociables d'un point de vue sémantique. Elles pourront tout aussi bien reprendre leur « indépendance temporelle » dès l'instant où leurs identités respectives divergeront.

83 Julia Blondeau, *Nachleben*. Pour soprano, baryton, percussions et électronique, 2014.

84 Julia Blondeau, *Théâtre d'ombres*. Pour 6 percussions, 2015.

*Organisation des voix et de leurs synchronicités*

Le second type d'organisation temporelle détermine l'agencement des différentes structures ou « voix » temporelles (du premier type défini ci-dessus). C'est elle qui traite donc du temps des synchronicités entre voix. C'est une super-structure de structures temporelles. Si les voix temporelles du premier type d'organisation sont souvent explicites, celles-ci le sont beaucoup moins et relèvent d'une organisation de plus haut niveau.

Un certain nombre de paramètres sont à prendre en compte pour l'organisation de ce « contrepoint de structures temporelles » et il faut pour chaque voix les mettre en vis-à-vis. Il s'agit notamment de mesurer les distances (d'identité musicale) entre chaque voix et d'organiser une certaine *harmonicité* ou *inharmonicité des temps*. Nous entendons par là qu'un ensemble de structures temporelles peut être plus ou moins « harmonique » (en référence à l'analyse spectrale), à savoir que les synchronicités se font plus ou moins périodiquement. Un exemple simple : les couches temporelles que l'on trouve dans les contrepoints de J.-S. Bach et dans lesquels plusieurs voix se dessinent, dans des temps différents mais dans des rapports très « harmoniques ». Cette conception se retrouve aussi bien entendue dans les principes de « spectres de formants » que Stockhausen élabore dans son article *Comment passe le temps*.

Deux structures temporelles peuvent être considérées comme inharmoniques entre elles si il n'existe que très peu, voir pas de points de synchronisation entre elles et que ceux-ci sont totalement apériodique. Dans ce cas il faudra prendre en considération un certain nombre d'éléments pouvant faciliter la perception des couches temporelles superposées et de cette apériodicité :

- degré de changements interne à chaque voix
- densité d'informations (horizontales)
- densité des couches
- degré d'hétérogénéité
- tension et attractivité
- ...

On connaît par ailleurs les écueils à éviter si l'on veut pouvoir percevoir deux couches temporelles de façon vraiment distincte. Il faut prêter attention aux divers effets de masquage et d'une façon générale, il faut procéder à une sorte « d'orchestration » des couches temporelles.

Les points de convergence, de synchronisation entre deux voix, que Marco Stroppa nomme des « pivots temporels<sup>85</sup> », les instants où s'établissent une simultanéité ne serait-ce que fugace, viennent marquer et structurer une durée d'échelle bien plus grande. Ces simultanéités coagulent pour faire émerger une forme du flux continu des durées. Il ne s'agit pas seulement de convergence vers une attaque ou une désinence commune, mais de points par lesquels une congruence plus ou moins complète peut avoir lieu. L'entremise mémorielle peut alors, par synthèse de ces points nodaux, nous placer d'une certaine manière devant le temps. C'est sur cet aspect que ce type d'organisation temporelle, qu'on pourrait placer à l'échelle mésoscopique, vient s'entrelacer à l'échelle temporelle supérieure : celle de la forme.

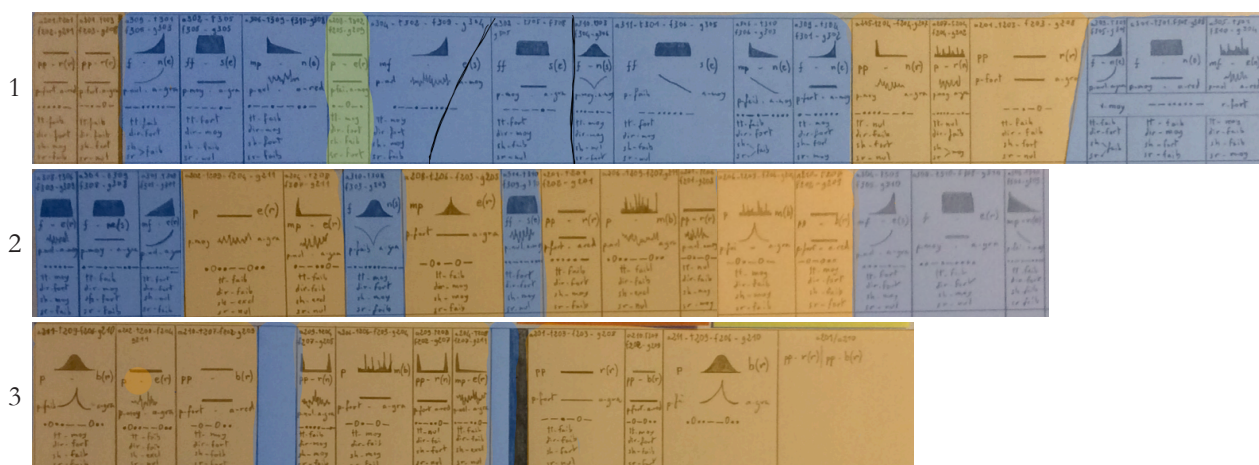
*Ces simultanés coagulent pour faire émerger une forme du flux continu des durées.*

85 Jacques Duthen et Marco Stroppa. *Une représentation de structures temporelles par synchronisation de pivots*. In Proceedings of Colloque « Musique et Assistance Informatique », pages 471–479, 1990.



*Organisation macroscopique des temps*

La troisième organisation est celle qui détermine l'échelle temporelle la plus grande, nous pourrions dire celle de la forme. C'est ici que les liaisons entre éléments deviennent parfois les plus implicites (tout du moins dans la partition) et viennent mettre en jeu des processus faisant directement appel à la mémoire. C'est dans cette organisation que se créent les trajectoires les plus éminentes qui feront que la forme ne se résumera pas à une suite de séquences choisies, peu ou prou, arbitrairement. C'est à cette échelle que devrait se produire une certaine *synthèse du temps*, qu'elle se produise par l'entremise mémorielle des différents moments qu'elle est alors en mesure de mettre en regard par *l'entendement* ou que ce soit par ce que Gilles Deleuze nomme « l'impression qualitative de l'imagination ».



Plan de *Sortir du noir*, pour violoncelle et électronique. Les couleurs correspondent à l'espace compositionnel utilisé.

« Mais à partir de l'impression qualitative de l'imagination, la mémoire reconstitue les cas particuliers comme distincts, les conservant dans "l'espace de temps" qui lui est propre. Le passé n'est plus alors le passé immédiat de la rétention, mais le passé réflexif de la représentation, la particularité réfléchie et reproduite. En corrélation, le futur cesse aussi d'être le futur immédiat de l'anticipation pour devenir le futur réflexif de la prévision, la généralité réfléchie de l'entendement (l'entendement proportionne l'attente de l'imagination au nombre de cas semblables distincts observés et rappelés). C'est dire que les synthèses actives de la mémoire et de l'entendement se superposent à la synthèse passive de l'imagination, et prennent appui sur elle.

[...]

Chaque coup, chaque ébranlement ou excitation, est logiquement indépendant de l'autre, mens momentanea. Mais nous les contractons en une impression qualitative interne, hors de tout souvenir ou calcul distinct, dans ce présent vivant, dans cette synthèse passive qu'est la durée. Puis nous les restituons dans un espace auxiliaire, dans un temps dérivé, où nous pouvons les reproduire, les réfléchir, les compter comme autant d'impressions-extérieures quantifiables. »<sup>86</sup>

86 Gilles Deleuze, *Différence et répétition*. Presses Universitaires de France, 2013. p. 98.

## De la structure interne au composé temporel

Il apparaît que ce que l'on désigne comme « qualité de temps » en composition est un composé de plusieurs facteurs non exclusivement temporels et que la spécification d'un tempo et de données rythmiques n'assure en aucun cas la pleine identification d'une qualité temporelle. Toute la conception de nos espaces compositionnels a été en partie réalisée suite à ce constat.

Un objet musical ponctuel (geste, fragment, son particulier...) en tant que tel, n'a que très rarement de qualité temporelle définie intrinsèquement. C'est la conjonction et la fabrication d'un composé de cet « objet musical » avec d'autres qui permet *in fine* d'arriver à la caractérisation d'une qualité de temps. Toute la composition se joue alors dans cette dialectique entre structure interne d'un objet et mise en temps de cette structure par la création de liaisons *sui generis* entre objets, mettant en place une structure temporelle d'échelle supérieure et un réseau de relations sous-jacentes.

La conjugaison d'éléments d'essences différentes et le choix de ces éléments particuliers sont résolument constitutifs d'une disposition temporelle qui, dans la composition de son développement, est à même de rendre manifeste une certaine qualité de temps. Il s'agit de choisir dans un premier temps un ensemble d'éléments singuliers qui seront mis en relation à l'intérieur d'un fragment, d'un motif, d'une phrase plus ou moins longue. Première disposition, premier montage statique. Ce sont des éléments atomiques qui déterminent les caractéristiques prégnantes susceptibles de définir une partie de l'identité musicale du passage : timbre, rythmicité, métrique, articulation, dynamique, enveloppe, données fréquentielles et harmoniques... Dans un second temps on définira à la fois l'évolution des éléments constitutifs et l'évolution des relations *entre* ces éléments. Quelle évolution du timbre, de la dynamique, quel lien y a-t-il entre les paramètres et comment évoluent-ils dans le temps ? Montage et disposition dynamique donc, à deux dimensions temporelles : simultanéité et durée.

Le tableau présenté ci-dessous résume cette idée. Divisé en deux paradigmes — durée et simultanéité — on peut y discerner le processus de composition utilisé avec les espaces compositionnels que nous avons préalablement présenté. La colonne de gauche (en bleu) dénote le choix des éléments de départ, puis la définition d'une cellule en conséquence. La colonne de droite est celle où l'on dispose, dans le temps et dans « l'espace de montage » des objets musicaux pour, implicitement, les lier entre eux. Ainsi, les deux colonnes de droite (en orange) ne définissent pas une cellule mais le déplacement vers une autre cellule, le déplacement d'un objet musical à un autre et si il y en a plusieurs, l'évolution de leurs interrelations.

1	1bis	2	2bis
<b>Choix d'éléments atomiques</b>	<b>Mise en relation, choix d'une disposition/configuration</b>	<b>Évolution des éléments atomiques</b>	<b>Évolution des structures</b>
timbre, métrique, éléments fréquentiels ou harmoniques, dynamiques, articulation particulière, rythmicité...	Construction d'un objet musical	changements d'état qualitatif ou quantitatif	modification des liaisons entre éléments atomiques dans le temps
densité d'informations degré d'hétérogénéité	<b>réseau, montage / disposition statique</b>	degré de changements tension et attractivité d'une configuration à l'autre	<b>remontage / redistribution dynamique</b>

Simultanéité

disposition statique, structure interne

Durée

champ de temps, structure temporelle

Notons ici que 1bis et 2bis sont respectivement induites du 1 et du 2 dans le processus compositionnel.

La notion de qualité de temps dépend ici, entre autres, de 3 dimensions : l'une verticale, statique, agaçant les paramètres entre eux et créant un premier réseau de relations, l'autre horizontale, dynamique, mettant en relation les différents objets et « fenêtres temporelles » créées par ceux-ci dans la durée, et une troisième « diagonale », liant les différentes couches temporelles entre elles en les synchronisant de manière plus ou moins importante et dessinant les différentes trajectoires sous-jacentes qui innervent la forme.

## Interdépendance des concepts de durée et de simultanéité

La caractérisation d'une qualité temporelle semble mettre en évidence un lien profond d'interdépendance entre des agencements appartenant au domaine de la simultanéité et des structures plus étendues relevant du domaine de la durée, tous deux étant très liés à un paradigme de montage (au sens d'un espace de *disposition*), l'un statique, l'autre dynamique. Cela est valable pour la composition d'une voix unique (structure interne et structure temporelle du 1er type défini plus haut) mais également pour les superpositions de voix et d'éventuelles couches temporelles (2ème type d'organisation temporelle).

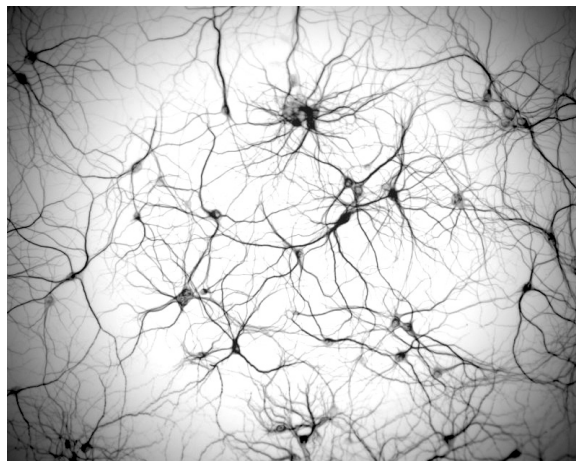
Durées et simultanéités sont dès lors *entrelacées* à diverses échelles : celle des structures internes faites d'éléments atomiques, celle des structures temporelles régissant l'évolution de ces structures internes, celle de l'agencement des différentes voix temporelles alors créées, celle de la disposition des points nodaux entre ces voix formés par les diverses synchronicités, et enfin celle de la forme.

Finalement, il nous semble qu'il y a un parallèle à faire avec le son en lui-même. On sait que l'identité d'un son ne peut se résumer à sa composition à un seul instant  $t$ . On sait à quel point l'attaque du son est déterminante pour son identification, mais aussi la manière dont il évolue et dont il « meurt ». À chaque instant, l'ensemble des harmoniques est constitutif mais l'identité du son ne se joue que dans l'évolution de cet ensemble, que dans les changements de relations entre les différents éléments. Un changement radical de relations peut ainsi faire passer un son d'un état harmonique à un état inharmonique. Si la configuration, le jeu des rapports entre chaque éléments à un instant  $t$  est important, leur évolution et les variations de leurs relations seront indispensables à l'édification de son identité. Alors seulement nous pouvons parler d'une composition de sa *qualité*.

*« Une dialectique du monteur, c'est à dire celui qui « dys-pose », séparant puis réajointant ses éléments au point de leur improbable rapport. [...] L'infamale relance des contradictions et, donc, la fatalité d'une non-synthèse. »<sup>87</sup>*

## Relations, intrications, entrelacs, devenirs

Une attaque-résonnance forte aura des fonctions tout à fait différentes si elle est superposée à un passage en itérations périodiques, un passage lisse et lent sans référence temporelle ou encore à l'origine d'un geste rapide. Dans le premier cas, elle viendra souligner un certain striage et marquer un cycle particulier. Dans le second cas, elle viendra fracturer une durée et marquera sans doute un événement fort dans la perception du temps du passage en question. Enfin, à l'origine d'un geste rapide, elle sera alors considérée comme élément déclencheur, créant pour la suite une attente implicite : si l'attaque apparaît, alors le geste doit suivre.

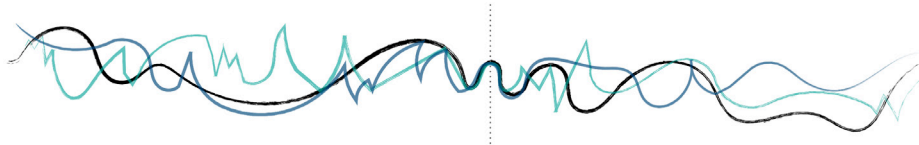


<sup>87</sup> Georges Didi-Huberman, *Atlas ou le gai savoir inquiet, L'œil de l'histoire 3*. Les Éditions de Minuit. p.13.

Au sein d'une même phrase musicale, d'un seul fragment, agissent et s'entrechoquent tous les éléments que constitue le langage musical. Lorsque l'on passe à une échelle un peu plus grande et qu'on distingue les enchevêtrements de lignes, on peut distinguer les moments où deux voix se croisent et créent par cela un événement qui viendra strier l'échelle temporelle supérieure. En préambule de ce chapitre, nous posons la question suivante : dans quelle mesure l'existence d'un continuum entre les notions de forme et de matériau serait-elle liée à la composition consciente d'entrelacs temporels ? Un continuum entre matériau et forme ne peut exister que si il y a un travail, une prise de conscience des relations existantes entre chaque échelle du temps musical. Si nous parlons d'entrelacs, c'est que l'image d'un enchevêtrement de fils, de courants, nous paraît correspondre à la réalité du discours musical qui, par sa puissance expressive, nous permet parfois d'aller au-delà de la morne linéarité du temps physique. Parfois un fil revient en arrière, il y a eu un appel à la mémoire qui nous permet de sauter d'un endroit à l'autre. Parfois un fil se casse, s'interrompt au milieu du labyrinthe de ses congénères. Parfois deux fils évolueront en parallèle, sans jamais se rencontrer. Parfois ils formeront une tresse ou fusionneront.

L'espace de pensée musicale dont nous avons parlé plus avant est pour le compositeur une sorte d'atlas, un lieu où faire monde. Il est un lieu de montage et de remontage mais aussi un lieu de tressage des différentes couches temporelles qui l'habitent. Par cela il est un lieu d'émergence de formes.

Ainsi, la forme n'est pas un modèle extrinsèque au matériau, elle en est un devenir.





## PARTIE II

# ORGANISER LE TEMPS





STAVROGUINE. – Dans l'Apocalypse, l'Ange jure que le temps n'existera plus.

KIRILOV. – Je sais. Ça y est net, vrai et précis. Lorsque l'homme tout entier aura atteint le bonheur, il n'y aura plus de temps, parce qu'il sera devenu inutile. Une idée très juste.

STAVROGUINE. – Mais où est-ce qu'on le cachera ?

KIRILOV. – On ne le cachera nulle part. Le temps n'est pas un objet, mais une idée. Il s'éteindra dans l'esprit.<sup>1</sup>

Les systèmes temps réel et les langages synchrones étendent très largement les possibilités d'écriture de l'électronique et ouvre ainsi une nouvelle expressivité en terme d'écriture du temps. Parmi ces nouvelles possibilités, nous étudierons, entre autres, la question des temps multiples et le principe de polyphonie temporelle, l'usage de processus informatiques permettant la mise en relation de processus musicaux complexes ou encore des stratégies de synchronisation liées à la question du phrasé en électronique. Le compositeur doit faire face à de nouveaux paradigmes : comment écrire le temps des actions et des processus électroniques et comment accorder ce temps à l'interprétation humaine lors d'une performance ? Il sera également question de notation, nous verrons que bon nombre de nouvelles questions se posent à ce sujet, autant du point de vue des compositeurs que des informaticiens. Il s'agira de discuter des différents types de notation pour la composition de musique électroacoustique et des possibilités actuelles — avec leurs avantages et leurs limites.

Ce travail s'appuie sur une collaboration étroite initiée sur le logiciel Antescofo, les langages synchrones et temporisés développés à l'Ircam dans le projet MuTant par Arshia Cont, Jean-Louis Giavitto, Philippe Cuvillier et José Echeveste, au sein de l'équipe Représentations Musicales.

## ANTESCOFO ET LES TEMPS MULTIPLES

Une introduction au langage Antescofo.....	69
Feuilletages et réseaux temporels .....	77
Les fenêtres temporelles.....	86
Stratégies de synchronisation : le temps du phrasé .....	89
« Mise en temps » d'un système par contraintes.....	97
Une taxinomie des échelles de temps .....	100

## NOTATION ET EXPRESSIVITÉ

La nécessité du symbole .....	102
Quelques réflexions sur différentes notations .....	109

## AU-DELÀ DE LA RÉTROACTION

Organiser le multiple pour organiser le temps ?.....	115
--	-----

---

1 Dostoïevski, *Les Possédés*.





# ANTESCOFO ET LES TEMPS MULTIPLES

Lorsque l'on parle de musique mixte et de temps réel, plusieurs types de temps entrent en jeu. Les deux plus évidents sont ceux de l'interprète et de la machine. Tout le jeu interprétatif consiste en de subtiles variations de nuances et de tempo permettant d'appuyer un phrasé, marquer le relief d'une phrase ou interpréter des variations locales de tempi inscrites sur la partition (ralentis, accélérés, etc.). La machine, quant à elle, évoluait à l'origine à tempo fixe, suivant le « temps du déclenchement » échantillonné par l'horloge physique. Mais grâce aux évolutions des suiveurs de partition et notamment à la machine d'écoute d'Antescofo, il est désormais possible d'accorder le tempo de la machine à celui de l'interprète. Dont acte, les temps de l'interprète et de la machine sont désormais coordonnés. Mais quelles nouveautés offrent alors un suiveur de partition en terme d'écriture du temps et notamment en terme de polyphonie temporelle ?

Le développement du suivi de partition et des langages tels que celui de la machine réactive d'Antescofo permet d'exprimer des relations temporelles de plus en plus fines. Si la composition de processus complexes enrichie l'écriture de l'électronique, la possibilité d'exprimer des relations entre processus ainsi que leurs liens avec l'interprète implique de nouvelles écritures du temps pour la musique mixte. Ces manières singulières d'exprimer les temporalités amènent avec elles des écritures radicalement différentes de celles, classiques, de la partition. Si certains inconvénients liés à l'écriture de programmes en « ligne de code » peuvent amener quelques problèmes de représentation, nous verrons que le langage Antescofo permet aujourd'hui une réelle écriture de l'électronique et des temps de celle-ci. Les apports d'un tel langage deviennent donc bien compositionnels dans la mesure où ils permettent d'exprimer une variété importante de temps musicaux, y compris dans leurs relations à un instrumentiste.

La partition est un support d'écriture et de pensée à travers lequel le compositeur organise des objets et des événements musicaux dans le temps. Ces objets datés dans un référentiel propre à la partition peuvent être composés dans des structures hiérarchiques, polyphoniques et séquentielles. Ils entretiennent le plus souvent entre eux des relations temporelles qui sont relatives les unes aux autres. Nous verrons tout au long de ce chapitre comment les exprimer dans le langage Antescofo.

## Une introduction au langage Antescofo

Antescofo est né des travaux de thèse d'Arshia Cont dans le contexte de création musicale de l'Ircam (notamment avec les compositeurs Marco Stroppa et Philippe Manoury). Depuis les articles séminaux de Roger Dannenberg et Barry Vercoe en 1984, les recherches sur le suivi de partition ont toujours constitué un domaine très actif. En 2007, Arshia Cont propose de coupler une machine d'écoute avec un langage permettant de décrire les séquences d'actions dans le temps de la partition. L'originalité de ce couplage est de permettre l'écriture du déroulement temporel de processus électroniques, en coordination avec un instrumentiste, sans pour autant figer le temps de ces processus. À l'image d'une partition classique, les relations temporelles des processus sont virtuelles et ne se réaliseront qu'au moment de la performance en fonction de l'interprétation humaine. L'objectif d'Antescofo était de permettre au compositeur d'exprimer des comportements électroniques dans un temps musical partagé avec celui d'un musicien. Nous ferons une présentation succincte du langage Antescofo pour pouvoir passer directement au sujet qui nous concerne. Le lecteur souhaitant s'informer précisément sur son fonctionnement se reportera aux thèses

de Arshia Cont, José Echeveste et Philippe Cuvillier. Nous mettons cependant à disposition du lecteur en annexe le User guide rédigé par nos soins durant cette thèse qui est une introduction au langage et qui donne quelques applications pratiques ainsi qu'un tutoriel orienté davantage composition et qui donne des exemples d'utilisation des processus et fonctions du langage.

Antescofo est un système musical interactif programmable. Bien qu'Antescofo soit souvent vu comme un suiveur de partition, avec une réponse séquentielle et un modèle interprète, le développement du langage a considérablement étendu ces caractéristiques, notamment grâce aux travaux et recherches de Jean-Louis Giavitto et José Echeveste. Le contexte de création musicale dans lequel s'est développé Antescofo a montré la nécessité de développer des outils permettant de décrire des scénarios interactifs complexes sortant du simple cadre du suivi partition. Ainsi le langage d'Antescofo est dédié à la création de structures temporelles complexes et il permet aujourd'hui de concevoir l'organisation temporelle d'applications pour la musique mixte, pour la musique improvisée, ou encore pour des installations interactives. La problématique du temps réel musical et ses critiques ont fortement influencé l'élaboration du langage temporel d'Antescofo. Le langage s'inspire de la relation complexe entre une partition et un interprète mais tente également de donner aux compositeurs un médium riche qui leur permette d'exprimer toutes les relations temporelles, y compris dans l'électronique seule qu'ils imaginent à l'intérieur de leurs partitions. Il y a alors possibilité d'une vraie dialectique entre couches temporelles : entre l'instrumentiste et l'électronique mais aussi entre toutes les couches temporelles qui peuvent exister au sein même de l'électronique. La réalisation de l'électronique n'est alors plus seulement un geste final d'assemblage mais un vrai instrument de composition et de conceptualisation.

Le système Antescofo vise à retrouver la puissance symbolique de la notation musicale classique dans le cadre nouveau de la musique mixte, où il est nécessaire à la fois de définir les parties électroniques et d'exprimer les interactions complexes entre les musiciens et ces processus lors du concert. Le compositeur écrit donc une partition « augmentée » comme un programme qui définit à la fois les événements du musicien qu'il faut reconnaître en temps réel (grâce à la machine d'écoute) et la spécification d'actions électroniques (grâce à la machine réactive). Pendant le concert, le moteur d'exécution du langage évalue la partition « augmentée » et contrôle l'évolution temporelle des processus électroniques en fonction de l'environnement musical grâce à la machine d'écoute artificielle. Celle-ci est capable de suivre le jeu du musicien en comparant le spectre du flux audio avec un spectre généré à partir des informations de la partition. Le modèle utilisé par la machine d'écoute repose sur des semi-chaînes de Markov cachées<sup>1</sup>. Ce modèle a été beaucoup amélioré par le travail de thèse de Philippe Cuvillier, notamment sur la prise en compte plus importante du temps.

## Actions et évènements

Une partition Antescofo correspond à la spécification de la partie instrumentale et des actions électroniques à réaliser lors de la performance musicale. Cette spécification est donnée en entrée de la machine d'écoute et du module réactif. Pendant le concert, le module réactif réagit aux données de la machine d'écoute (position courante du musicien dans la partition + estimation du tempo) et de l'environnement extérieur pour exécuter des actions d'accompagnement. Un des principaux atouts d'Antescofo réside dans le fait qu'un compositeur peut exprimer la temporalité des événements et des actions relativement :

- aux événements du musiciens
- au tempo du musicien ou à un tempo calculé
- à un mixte des deux précédents

---

<sup>1</sup> Yann Guédon. *Hidden hybrid markov/semi-markov chains*. Computational statistics & Data analysis, 49(3) :663–688, 2005.

Au sein d'une partition, le compositeur peut par exemple décider de déclencher des actions au moment de la détection d'un événement musical, après un certain délai évalué à la volée, exprimé en pulsations ou en secondes. Il peut également grouper des actions de façon hiérarchique, les lancer en parallèle, les itérer et définir des comportements temporels.

La partie instrumentale est décrite à l'aide de mots clés correspondants à des événements musicaux : notes, accords, trilles, glissandi, etc. Dans l'exemple suivant deux événements sont spécifiés :

```
NOTE C4 1 e1
CHORD (C4 F#4) 1/2 e2
```

Ils correspondent à la partition du musicien et devront être reconnus par le système lors sa prestation. Le premier est un do C4, qui dure un temps, le deuxième est un accord de do et fa# (C4 F#4) qui dure 1/2 temps. Des labels optionnels, e1 et e2 permettent de se référer à ces événements ailleurs dans la partition.

La syntaxe du langage permet de définir 2 catégories d'actions. On peut leur associer un label dont la portée est toujours globale. Une action atomique correspond à une instruction élémentaire qui s'exécute en temps 0 (hypothèse synchrone) et une action composée structure dans le temps un ensemble d'actions. Les actions composées peuvent s'imbriquer les unes dans les autres. La durée d'une action composée correspond à l'intervalle de temps qui sépare son déclenchement de l'exécution de la dernière action atomique dans la hiérarchie. Chaque action peut être précédée d'une expression optionnelle qui correspond au délai séparant son exécution de la détection de l'évènement précédent ou l'exécution de l'action précédente.

```
NOTE C4 2
1/2 action1
3 action2
NOTE D5 1
```

## Actions composées

Les actions composées sont des structures de contrôle permettant de séquencer d'autres actions (atomiques ou composées) dans le temps ; elles sont simples, répétées ou continues. Ces actions permettent au compositeur de construire facilement des hiérarchies imbriquées et organisées en fonction de leur comportement. Des attributs permettent d'associer à une action composée : un nom, un tempo local, une stratégie de synchronisation et de rattrapage d'erreurs. S'ils ne sont pas spécifiquement définis, les attributs sont hérités de l'action composée englobante.

**Groupe.** La construction `group` séquence au sein d'un même bloc des actions partageant des propriétés communes de tempo, de synchronisation, de gestion des erreurs, etc. Elle facilite l'écriture de phrases indépendantes pour former une polyphonie, sans devoir calculer l'entrelacement entre les actions. Il devient ainsi simple de signifier la co-existence de phrases distinctes ou la spécification d'un ensemble de paramètres correspondant aux caractéristiques d'une même entité musicale.

```
GROUP { actions* }
```

**Boucle.** La construction `loop` est similaire à `group` mais la séquence d'actions est itérée après un certain délai (`period` dans l'exemple suivant). Le délai, évalué à chaque itération, correspond à la période de la boucle quand il est constant. Un phénomène de tuilage peut se produire si la durée du bloc d'actions est supérieure à la période.

Pour une boucle, les clauses de terminaison sont évaluées à chaque début d'itération. La période d'une itération  $n$  est évaluée au début de l'itération  $(n - 1)$ .

```
LOOP period { actions* } until expression
```

**Interpolation.** La construction `curve` effectue des calculs d'interpolation de flottants. L'utilisateur peut choisir le type et le pas (`@grain`) d'interpolation. Il peut spécifier les valeurs de références (les paliers), la durée entre chaque palier, les variables affectées et l'action à exécuter à chaque pas d'interpolation. Le pas et la durée entre les paliers peuvent être spécifiés en temps relatif ou en temps absolu. Cette construction permet de contrôler l'évolution de paramètres de processus musicaux de façon « quasi continue ». L'utilisation la plus classique étant le contrôle d'un paramètre d'amplitude par une enveloppe réalisée par la `curve`.

```
curve grain action
  [var*]
  {
    ( delay          [expression*] interpolation type
      delay          [expression*] interpolation type ) *
      [expression*]
  }
```

## Processus

Les processus sont des actions créées dynamiquement. Un processus est spécifié par la construction `@proc_def`. Tous les noms de processus sont introduits par `::`. Après sa définition, un processus `::P` peut être appelé avec des arguments. Cet appel exécute alors une instance de ce processus sous la forme d'un groupe d'actions (plusieurs instances d'un même processus peuvent s'exécuter en parallèle).

```
@proc_def ::echo($pitch, $d)
{
  $d/3 loop $d/3
  {
    play $pitch
  } during [2#]
}
NOTE C4 2
::echo(«C4», 2)
NOTE D3 1
::echo(«D3», 1)
```

Dans l'exemple précédent le processus `::echo` répète deux fois la note qui l'a déclenché, réparties pendant la durée de celle-ci. On utilise pour cela une boucle, lancée après un délai de  $\$d/3$ , de période  $\$d/3$  qui envoie le message `play $pitch` à chaque tour de boucle, au nombre de deux (`during[2#]`).

Nous listons ici quelques propriétés spécifiques aux processus :

- un processus peut aussi être appelé dans une expression. L'instance du processus est alors lancée et la valeur retournée correspond à la coroutine réalisant le corps du processus ;
- les processus peuvent être récursifs ;
- leurs arguments sont calculés au moment du lancement du processus ;
- il est possible de terminer tous les processus d'un nom donné ou d'une instance spécifique :

```
$p1 := ::P()
$p2 := ::P()
$p3 := ::P()
10.0 abort $p1 ; terminaison du processus $p1
1 abort ::P ; terminaison de tous les processus ::P
```

## Les motifs temporels

La structure `whenever` permet de réagir à un événement. Dans son utilisation classique, le compositeur spécifie à l'aide d'une combinaison de variables une condition qui correspond à l'événement déclencheur lorsqu'elle est vérifiée. Les motifs temporels facilitent la spécification d'événements complexes comme par exemple une succession d'événements avec des contraintes temporelles particulières. Les événements peuvent être ponctuels (`EVENT`) mais peuvent aussi avoir une durée (`STATE`). L'exemple ci-dessous montre comment définir l'événement complexe `pattern::P` correspondant à la configuration suivante : la variable `$x` est supérieure à un seuil égal à 10 pendant 0.5s puis la variable `$y` prend la valeur 1 avant qu'une pulsation ne soit passée. On utilise ensuite ce `pattern` dans la spécification d'une condition pour associer une réaction à cet événement.

```
@pattern_def pattern::P
{
    STATE $x where($x>10) during 0.5s
    before 1
    EVENT $y where($y=1)
}
...
whenever (pattern::P) { action* }
```

## Tempo et pulsation

Dans le langage d'Antescofo un délai est une expression et peut être spécifié en pulsations (par rapport à un tempo) ou en temps physique (en secondes ou millisecondes). Le tempo, exprimé en nombre de pulsations par minute, spécifie « la vitesse d'écoulement du temps de la partition ». Un tempo peut être associé à une action composée, il permet alors d'interpréter les délais exprimés en pulsation. Par défaut, le tempo réfère au tempo estimé en temps réel à partir des événements du musicien reconnus par la machine d'écoute.

Dans Antescofo les compositeurs peuvent introduire leurs propres référentiels en spécifiant un tempo sous la forme d'un attribut pour toute action composée (groupe, boucle, interpolation).

```
GROUP @tempo := $tempCompositeur
{
    ...
}
```

Ce tempo est une expression arbitraire réévaluée chaque fois que nécessaire pour ajuster la manière dont on « compte » le temps dynamiquement. Les actions filles héritent du tempo de l'action englobante à moins qu'un nouveau tempo ne leur soit attribué. Nous verrons un peu plus tard les possibilités de modifications dynamiques du tempo ainsi que les notions de phrasé ou de fenêtre temporelle qu'elles peuvent apporter.

## Variables utilisateurs

Comme dans les langages impératifs, les variables Antescofo correspondent à des zones mémoires dont le contenu peut changer au cours de l'exécution. Elles sont par défaut globales, accessibles en lecture et écriture n'importe où dans la partition. On peut cependant spécifier qu'une variable est locale à une action composée, limitant ainsi sa portée et sa durée de vie. On peut ainsi utiliser une variable locale à un processus de façon à ce qu'elle soit circonscrite à cet unique processus. Un identificateur de variable débute par le caractère `§`.

Les variables peuvent être mises à jour explicitement dans la partition par l'affectation ou via un mécanisme permettant de mettre à jour les variables depuis l'environnement extérieur. Ainsi Antescofo peut réagir non seulement aux événements musicaux notifiés par la machine d'écoute mais aussi à toutes sortes d'événements en provenance de l'environnement.

## Abstractions

Le tableau suivant<sup>2</sup> indique les différentes abstractions du langage Antescofo ainsi que leur description et usage. Nous verrons tout au long de ce chapitre dans quels contextes ces abstractions sont utilisées et comment les combiner.

<i>Abstractions</i>	<i>Description et usage</i>
événement	décrit un événement joué par le musicien reconnaissable par la machine d'écoute
action atomique	calcul ou message de durée nulle
action composée :	contrôle l'ordre et la temporalité des exécutions
- group	séquence des actions dans le temps (phrase musicale)
- loop	itération une séquence d'actions
- curve	interpolation de paramètres continus dans le temps (geste musical)
processus	permet d'abstraire des séquences et de les rejouer dynamiquement à la demande
label	permet de désigner un événement musical ou une action
nom de variable	permet l'accès à une donnée
nom de fonction	dénote une valeur fonctionnelle
structure de donnée :	permet d'organiser les données sous forme de tableau
- tab	tableau
- map	dictionnaire
- NIM	représentation symbolique d'une fonction interpolée
liaison lexical des variables	fournit l'accès à la valeur dans le contexte de définition
liaisons dynamique des attributs temporels	héritage des attributs temporels du contexte d'appel
attribut de synchronisation	stratégie d'alignement temporel d'une séquence d'actions avec la performance du musicien
attribut de rattrapage d'erreur	gestion des événements manqués ou non-reconnus
délai	durée temporelle permettant de spécifier la date d'une action relativement à l'action ou l'événement qui la déclenche
tempo	spécification de l'unité de temps utilisée pour mesurer les délais

<sup>2</sup> José Echeveste, *Un langage de programmation pour composer l'interaction musicale*. Thèse de doctorat. 2015.

## Du déclenchement à la synchronisation

Le langage Antescofo permet de soutenir deux visions complémentaires de la musique mixte, dans le rapport entre électronique et jeu instrumental. Historiquement, le suivi de partition a largement mis en avant un paradigme de déclenchement, à savoir qu'à un événement instrumental reconnu, on associait le lancement d'une ou plusieurs actions dont l'évolution dans le temps restait indépendante du jeu instrumental. L'extrait d'*Anthèmes 2*, de Pierre Boulez, présenté ci-dessous, est très représentatif de cette approche. À l'époque de la création, le suivi de partition n'étant pas suffisamment fiable, le déclenchement est assuré par une pédale MIDI et l'on peut aisément voir sur la partition que l'électronique est pensée comme une suite d'actions déclenchées de manière discrète.

Régie informatique

**Anthèmes 2**  
pour violon et dispositif électronique (1997)

**Pierre Boulez**  
(\*1925)

Libre brusque (♩ = 92) (♩ = 92) *rall.* (♩ = 66)

Violon *f* *fff* *mf* *ff* batt. (archet normal)

Spatialization F -11/-18/-18/2.0

Inf. Rev. reverb. time: 60°

Spatialization F -11/-18/-18/2.0

Sampl. IR MIDI: 93 92 88 84 82 80 79 77 75 74 reverb. time: 60°

Spatialization F -11/-18/-18/2.0

Sampler *pizz.* *pizz.* = 93 msec. = 93 msec.

MIDI: [74 73 70 69 68 67 66 65] [74 73] [89 70 73 74] [74 75 72 69 66] [87 68 71 72 73 74] [83 64 67 68 69 70 71 74] [74 73 72 71 70 67 66]

Spatialization MR -4/-12/-24/2.0

Freq. Shift.

Spatialization

Les avancées en terme de suivi, de rattrapage d'erreurs et de stratégies de synchronisation (nous reviendrons sur ce point plus tard) permettent aujourd'hui de pouvoir penser l'électronique comme possible composée d'éléments continus qui vont pouvoir évoluer non seulement parallèlement mais aussi de manière synchrone à l'interprète instrumentiste. La partition présentée ci-dessous est un extrait de *Nachleben*, composée au commencement de cette thèse et qui utilisait notamment les possibilités d'écriture de courbes, et de processus continus d'Antescofo parallèlement à des éléments plus classiques de déclenchement d'actions sporadiques. La partie électronique est divisée en deux voix, une soliste (pour un totem de haut-parleurs sur scène) et un « tutti » (pour les haut-parleurs placés autour du public). On peut distinguer les différents types d'écriture possibles sur un spectre allant d'une écriture événementielle à une écriture de processus continus.

Nous verrons tout au long du chapitre comment il est possible de spécifier des types particuliers de synchronisation pour obtenir une plus ou moins grande souplesse de l'électronique. Nous verrons les différentes manières de composer différentes couches temporelles et d'organiser leur intrication.



ORGANISER LE TEMPS : ANTESCOFO ET LES TEMPS MULTIPLES

**S.** 31 *f* *mf* (de l'intérieur) *mp* *mf*  
 Le cours de l'expérience a chuté. [a][e] ad al-tra lu - ce

**B.** *gliss* *ff* *mf*  
 [e] [a] in-cer-ta al - ba [lu]

**P.** *bol* *mf* *f* *sfz* *ff* *sfz* *ff* *sfz*  
 (face au public, se lever brusquement) (s'asseoir) *P* *D+G* *D-G*  
 Die Erfahrung ist im Kurse Gefallen!

**Electro Solo** *f*

**Electro Tutti** *mf* *ff*

RT-Tempo 60 30 RC

50 50 35 35

## Feuilletages et réseaux temporels

Les diverses manières de parler du temps entraînent avec elles des utilisations musicales particulières. La possibilité d'exprimer notamment le temps en valeurs absolues ou relatives implique la spécification d'une certaine souplesse ou au contraire d'une raideur vis-à-vis de l'interprète. Mais, plus que cela, il existe des actions (boucles, processus récursifs ou non...) dont la nature même change la manière de concevoir le temps lorsqu'on les utilise.

L'utilisation de boucles (dont le tempo peut être contrôlé depuis l'extérieur) dans lesquelles la manière de sortir de celles-ci peut être spécifiée et dont les variables peuvent évoluer dynamiquement au cours du temps amène une notion hybride de temps cyclique séquencé permettant d'échapper à l'écriture, souvent triviale, de boucles identiques répétées *ad nauseam*. Le temps y est certes cyclique, mais l'altération progressive des répétitions entraîne avec elle la mutation, voir l'implosion, de la périodicité.

On pourra obtenir une large palette de « rapports à l'extérieur » allant du plus rigide, avançant en cercle fermé dans son propre tempo, au plus souple, ajustant en permanence son tempo et recevant des variables extérieures venant constamment altérer son état.

*Le temps y est certes cyclique, mais l'altération progressive des répétitions entraîne avec elle la mutation, voir l'implosion, de la périodicité.*

Les groupes d'actions peuvent avoir cette même palette de dépendances temporelles mais évoluent en revanche dans un temps rigoureusement séquentiel. On verra par la suite que le type de synchronisation d'un groupe influera largement la souplesse de son temps. On jouera donc à la fois sur les paramètres de tempo et de synchronisation pour arriver à une souplesse de jeu définie spécialement pour des raisons musicales.

Un certain nombre d'actions conditionnelles (comme le `if` ou le `whenever` en Antescofo) soulèvent la question du « hors-temps », leurs actions respectives n'étant déclenchées que dans la mesure où une certaine condition serait respectée. Cela engendre un type de composition particulier qui comme on peut l'imaginer se prête particulièrement bien aux œuvres ouvertes, mais pas seulement. Nous le verrons plus loin, ces actions impliquent une sorte de composition d'ordre supérieur, se plaçant alors en « surimpression » de la partition. Elles impliquent une certaine logique temporelle qui suppose une écriture de haut-niveau (au sens informatique du terme) et la désignation d'éléments d'ordre éminemment formel (au sens musical).

### Types temporels

Chaque action électronique peut se subdiviser en un ensemble de composantes qui peuvent impliquer des types de temps différents :

- lancement d'une séquence d'actions atomiques
  - temps de l'évènement organisé dans le temps de manière séquentielle par synchronisation avec le musicien
- lancement de cycles indépendants ou non (loops)
  - temps cyclique continu ou discontinu
- lancement de processus indépendants ou non
  - fonction atemporelle « mise en temps » par l'interprète.
- lancement de lignes/courbes continues
  - temps continu lié à l'interprète
  - temps continu lié à l'interprète mais ayant son tempo propre (avec ou sans évolutions)

Le code suivant est un exemple relativement classique de séquence d'actions atomiques. La particularité réside dans le fait que les actions lancées sont des processus « mis en temps » au moment de leur lancement. Ici un processus lance un moteur de synthèse (`::ASC0toCS_percu`) pendant qu'un autre assigne une source de spatialisation à ce moteur (`obj::spat`).

```
GROUP
{
  ::ASC0toCS_percu(5,2,0.05,84.,0.002,0.2,7)
  obj::spat("perc7mes8",1,"csound7","vbap",::spat_traj_pos,[30.,4.,7.])
  1/10 ::ASC0toCS_percu(15,2,0.08,84.,0.003,1.6,8)
  obj::spat("perc8mes8",1,"csound8","vbap",::spat_traj_pos,[220.,4.,7.])
  1/10 ::ASC0toCS_percu(13,2,0.1,84.,0.004,0.4,9)
  obj::spat("perc9mes8",1,"csound9","vbap",::spat_traj_pos,[90.,4.,7.])
  1/10 ::ASC0toCS_percu(17,2,0.15,84.,0.004,0.6,1)
  obj::spat("perc1mes8",1,"csound1","vbap",::spat_traj_pos,[315.,4.,7.])
  1/10 ::ASC0toCS_percu(18,2,0.2,84.,0.005,1.6,2)
  obj::spat("perc2mes8",1,"csound2","vbap",::spat_traj_pos,[180.,4.,7.])
}
```

L'exemple ci-dessous lance une boucle « pizzic » qui est stoppée au bout de 8 pulsations. À chaque nouvelle itération de la boucle, le type de synthèse (processus `::ASC0toCS_percu`) change (`$randInst`) et la variable d'amplitude suit la valeur donnée par une curve extérieure (`$amperc`). Cet exemple aurait pu s'écrire de façon beaucoup moins expressive (au sens informatique) en écrivant chaque itération de façon séquentielle comme dans l'exemple précédent et en spécifiant les valeurs de chaque paramètre. La notation dénote donc ici le procédé musical.

```
obj::spat("AleaPerc",18,"csound2","vbap",::spat_traj_fun,[::lissaj2inv,8,30,0,3.])
@local $randInst,$amperc
$amperc := 0.4
loop pizzic 1/8 ; @tempo := 76
{
  $randInst := (@rand_int(16))+3
  ::ASC0toCS_percu($randInst,2,$amperc,84.,0.003,1.5,2)
}
curve percamp @grain := $grain_curve
{ $amperc
  { 0.4 }
  8 { 0. }
}
}
8 abort pizzic
```

L'exemple suivant utilise toujours une boucle, mais celle-ci contient une enveloppe « continue », elle-même modulée par une autre boucle externe. Notons l'attribut `@abort`, très utile puisqu'il permet de spécifier la façon dont la boucle se terminera. Cet attribut permet de travailler avec des boucles dont on ne spécifie pas de condition terminale mais qu'on souhaite arrêter suivant des paramètres extérieurs dont on ne connaît pas la temporalité. Cette boucle sera dans tous les cas terminée de la manière dont on l'aura spécifié dans le corps du `@abort`. Le tempo de la boucle est, lui, spécifié de l'extérieur par la variable `$tloop1` qui est envoyée depuis une curve qui produit ici un ralentissement important.

On voit ici, sur un exemple relativement simple, toutes les subtilités possibles de combinaisons des diverses possibilités de jeu entre variables et entre les différents types temporels. On voit par ailleurs l'utilisation de 2 types de spécifications temporelles correspondant à 2 utilisations différentes. Ainsi, les grains de curves sont ici spécifiés en secondes, ce qui s'approche d'une notion plus proche du signal audio ; on pourrait transcrire ce grain par un taux d'échantillonnage. En revanche, les temps spécifiés à l'intérieur des curves sont spécifiés en pulsation. Cela permet d'exprimer des durées en terme plus musical.

```

1 loop SurAmp 7
{
  curve ampg13 @grain := 0.04s
  { $ampg13
    { { 1. }
      4 { 1.2 }
      3 { 1. }
    }
  }
}

curve tloop1 @grain := 0.04s
{ $tloop1
  { { 56. }
    25 { 10. }
  }
}

loop ampg1s 1 @tempo := $tloop1
@abort {
  curve ampg1s @Grain := 0.02s, @Action := CS_solo_lignes c g1.amp $ampg1s
  { $ampg1s
    { { $ampg1loc }
      2 { 0.0 }
    }
  }
}

{
  curve ampg1s @Grain := 0.02s, @Action := CS_solo_lignes c g1.amp $ampg1s
  { $ampg1s
    { { 0.1 } @type "quad_in_out"
      1/2 { (0.2*$ampg13) } @type "quad_in_out"
      1/2 { 0.1 }
    }
  }
  $ampg1loc := $ampg1s
}

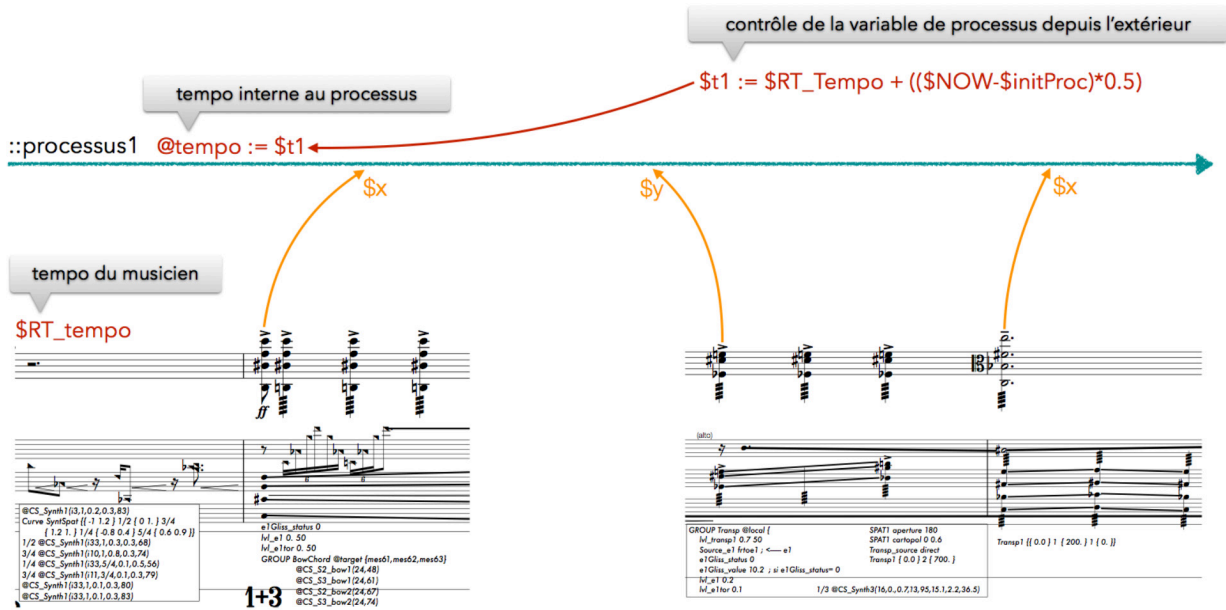
```

Les différents types temporels (séquentiel, cyclique, « atemporel », continu) vont donc nous permettre d’exprimer toute la diversité des rapports qui peuvent exister en composition lorsqu’on souhaite écrire des « polyphonies temporelles » plus ou moins complexe. Nous en verrons au cours de ce chapitre plusieurs exemples qui mettront en évidence l’importance de question comme celle de l’*expressivité* lorsqu’il s’agit de composer avec l’électronique.

## Polyphonie, liaisons et interactions entre couches temporelles

Notre recherche s’est orienté sur une notion de polyphonie temporelle et sur les diverses manières d’organiser le temps, que ce soit du point de vue de l’interprète, de l’électronique ou de leurs relations plus ou moins interdépendantes. Le temps de l’interprète est dénoté par une variable `$RT_TEMPO` interne à Antescofo. Cette variable est utilisée par défaut mais on peut décider, pour chaque action, d’en utiliser d’autres définies statiquement ou dynamiquement. Le tempo d’un groupe ou d’un processus peut alors venir d’un calcul, d’une variable externe ou d’une modulation de la variable `$RT_TEMPO`. On peut alors créer des réseaux de dépendances entre processus selon leurs différents référentiels de temps.

L’exemple ci-dessous montre un processus dont les variables `$x` et `$y` correspondent à des observations sur l’interprète au cours du temps (hauteurs, mode de jeu, amplitude...) et dont le tempo, dès l’instant où le processus s’enclenche, démarrera au tempo du musicien pour augmenter au fur et à mesure de l’avancement du temps (spécifié par l’expression : `$RT_TEMPO+(( $NOW-$initProc)*0.5)`). Les variables `$x` et `$y` viennent stopper, pour une période de temps définie, cette altération du tempo du processus.



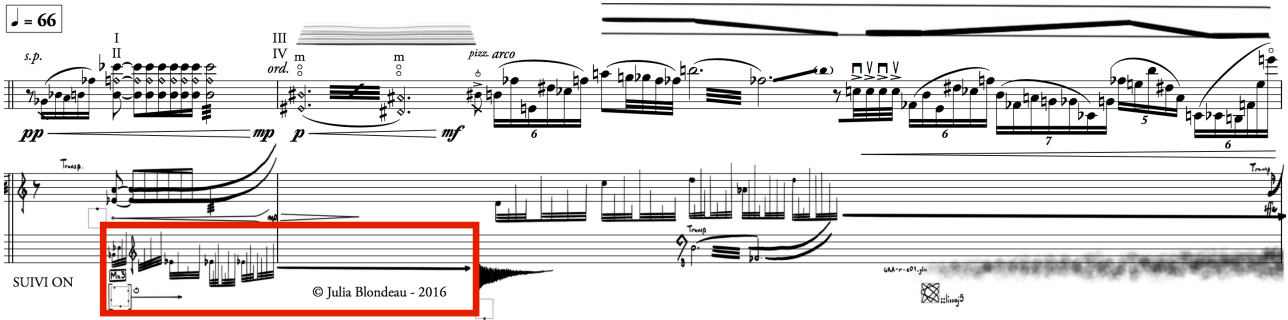
Cet exemple sommaire montre les possibilités d'interactions qui peuvent être mise-en-œuvre dans la composition de processus évoluant parallèlement à l'interprète mais prenant en compte son avancement temporel et s'altérant par son intervention sur certaines variables. Nous pourrions donc imaginer des entrelacs temporels dont les maillages seront diversement interconnectés entre eux et à l'intérieur desquels l'interprète agira et évoluera de manière souple.

L'écriture de variations de tempo est ici assez simple. Il est possible de faire coexister des voix dont les temps évoluent plus ou moins indépendamment les uns des autres mais également de composer des variations temporelles fines à l'intérieur d'une même phrase musicale (nous reviendrons sur cette notion un peu plus tard). Le compositeur se heurte cependant à des difficultés d'ordre notationnel qui mettent en évidence des problèmes de sémantique musicale. Dans le cas de variations de tempo complexes et agissant sur plusieurs groupes en parallèle, il devient difficile de se figurer de façon rapide et claire les points de synchronisation communs aux autres couches temporelles (celle de l'interprète ou celles des autres strates électroniques). Les temps de chaque groupe étant tous écrits en temps relatif (ce qui est extrêmement pratique durant la composition) et leurs évolutions de tempo déterminées par une courbe extérieure, leur déclaration se fait alors de façon totalement symbolique et la représentation graphique actuelle ne permet pas une visualisation réelle des évolutions temporelles de chaque groupe. La puissance du langage symbolique est ici contrebalancée par la nécessité de pouvoir se faire une représentation mentale claire du contrepoint temporel en cours de conception. Ce problème devra être résolu si l'on veut atteindre un jour une souplesse d'écriture qui nous permette d'agir en pleine conscience des corrélations temporelles existantes entre deux ou plusieurs strates temporelles. Nous pourrions alors tendre vers une véritable écriture temporelle contrapuntique.

La page qui suit donne un exemple<sup>3</sup> dans lequel la voix du violoncelle est complétée par une voix électronique qui a pour particularité de générer une séquence de hauteurs (attaques de synthèse assez saillantes) selon certaines contraintes et dont le tempo tourne autour du tempo de l'instrumentiste. Le fait de « tourner autour » du tempo instrumental est doublé par le fait que la séquence générée vient se coller régulièrement aux hauteurs jouées par l'instrumentiste. Les notes pivots deviennent alors des pivots de tempo et c'est une sorte de tresse qui se tisse autour du jeu du violoncelle.

3 Il s'agit ici de la mesure 8 de *Sortir du noir*, pour violoncelle et électronique.

# ORGANISER LE TEMPS : ANTESCOFO ET LES TEMPS MULTIPLES



```

TRILL ((Eb4 A4)) 7/2 ; harmonics

GROUP continuum @target [1/2]
{
  @local $cpt
  SPAT-synt1 xy -0.8 1
  SPAT-synt2 xy 0.9 -0.4
  SPAT-synt3 xy -1 -0.3
  SPAT-synt4 xy 1 0.8
  SPAT-synt5 xy -0. -1

  $tempContinuum := 66
  $stabSP := [1,2,3,4,5] ; sources synthèse
  $cpt := 0

  curve @grain := 0.05s
  { $ampLoop
    {
      { 0.5 } @type "quad"
      3/2 { 3. } @type "cubic_out"
      5 { 5. } @type "quad_out"
      1/2 { 9. }
      7/2 { 6.5 }
      3 { 3. }
      3 { 9.5 }
      4 { 9.5 }
      1 { 10.5 }
      2 { 8.5 }
      1 { 9. }
      8 { 12. }
      1 { 0.5 }
    }
  }

  curve @grain := 0.1s
  { $tempContinuum
    {
      { 86 }
      3/2 { 92 }
      5 { $RT_TEMPO }
      7 { 92 }
      3 { $RT_TEMPO }
      4 { 92. }
      1 { 82. }
      2 { 92 }
      1 { $RT_TEMPO }
    }
  }

  $stabGener := 1 ; pour envoyer un tableau d'intervalles généré sous contraintes
  $freqs := @HTgenerInt_asc($M3, 46, "ia3m", 42, 100, 7)
  $freqsB := [82,82.5,82,81.8,82,82.4,81,82] ; polarisation

  loop continuumPoint 1 @exclusive @tempo := $tempContinuum
  {
    $cpt := $cpt+1
    ::ASCtoCS_SYNTH_L("g1",1/3,($freqs[0]),$ampLoop,2,0.,0.,0.2,5,$stabSP[0])
    1/8 ::ASCtoCS_SYNTH_L("g2",1/3,($freqs[1]),$ampLoop,4,0.,0.,0.2,5,$stabSP[1])
    1/8 ::ASCtoCS_SYNTH_L("g3",1/3,($freqs[2]),$ampLoop,7,0.,0.,0.2,5,$stabSP[2])
    1/8 ::ASCtoCS_SYNTH_L("g4",1/3,($freqs[3]),$ampLoop,3,0.,0.,0.2,5,$stabSP[3])
    1/8 ::ASCtoCS_SYNTH_L("g5",1/3,($freqs[4]),$ampLoop,5,0.,0.,0.2,5,$stabSP[4])
    1/8 ::ASCtoCS_SYNTH_L("g6",1/3,($freqs[5]),$ampLoop,1,0.,0.,0.2,5,$stabSP[0])
    1/8 ::ASCtoCS_SYNTH_L("g7",1/3,($freqs[6]),$ampLoop,6,0.,0.,0.2,5,$stabSP[3])
    1/8 ::ASCtoCS_SYNTH_L("g8",1/3,($freqs[7]),$ampLoop,2,0.,0.,0.2,5,$stabSP[2])

    $stabSP := @rotate([1,2,3,4,5],$cpt)
    if ($cpt>=4) {$cpt:=0}

    if ($PITCHE==0)
    { $PITCHE := ($freqs[1]) }
    else { $PITCHE := $PITCHE }

    if ($stabGener == 1)
    { $freqs := @HTgenerInt($M3, (@abs($PITCHE)/100), @num2int((@car(@rotate($freqs,1))-@car(@rotate($freqs,2))))), ((@abs($PITCHE)/100)-20), ((@abs($PITCHE)/100)+20), 7) }
    else { $freqs := $freqsB }
  }
}

```

Contrôle du tempo de la boucle

Corps de la boucle  
avec génération de séquence (tableau de hauteurs) et itérations de processus selon la génération et le contrôle d'amplitude.

Si on regarde les valeurs de la curve contrôlant les variations du tempo de la boucle « continuum-

Point », elles sont assez arbitraires, dans le sens où elles ont été choisies de façon complètement empirique, à l'écoute du résultat et par essai-correction. Nous verrons dans un chapitre ultérieur un moyen de dénoter une variation de tempo de manière symbolique, de façon à ne pas avoir à entrer de valeurs qui ne font pas véritablement sens mais qui « collent » au résultat attendu par le compositeur.

### Entrelacs électroniques

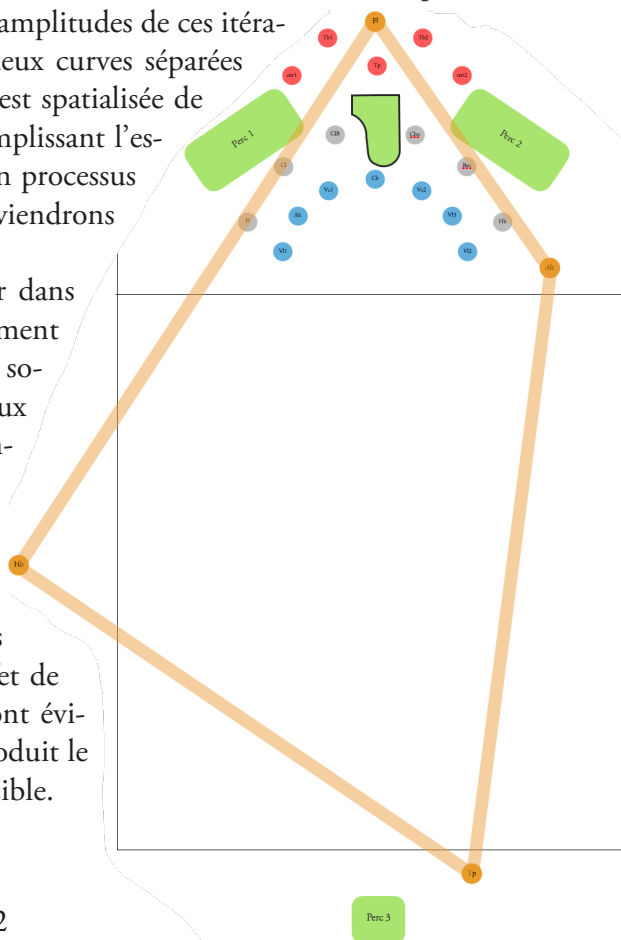
L'exemple suivant est un autre type d'entrelacement temporel (le code du processus correspondant est en page suivante). Dans celui-ci — tiré de la pièce *Namenlosen* — les 4 solistes évoluent indépendamment les uns des autres, avec un tempo propre à chacun. Leur éloignement « géographique » — ils sont placés aux 4 coins de la salle de concert (en orange dans le plan présenté en bas de page : la flûte en haut, l'alto à droite de la scène, la trompette en bas, décentrée sur la droite et le hautbois à gauche du public) — permet de dissocier les différentes voix lorsque celles-ci se recouvrent sur le même spectre.

Extrait de *Namenlosen*, partie IV : 4 voix temporelles entrelacées par l'électronique

La flûte joue à un tempo de 80, l'alto est à 60, la trompette à 48 et le hautbois à 72. L'électronique va ici jouer le rôle d'entrelacs temporel. Une séquence d'itérations est lancée dans un processus dont le tempo passera par chacun des tempi des solistes. Les amplitudes de ces itérations sont gérées sur l'ensemble du processus par deux courbes séparées (pour les itérations graves et aiguës). L'électronique est spatialisée de façon à « voyager » d'un soliste à l'autre, tout en remplissant l'espace grâce à des courbes de Lissajous générées par un processus Antescofo dédié à la spatialisation sur lequel nous reviendrons par la suite (`obj::spat`).

L'idée de ce passage était de plonger l'auditeur dans plusieurs flux distincts — discernables individuellement pour peu qu'on focalise son attention sur chacun des solistes — qui cependant soient liés entre eux par un flux tournoyant qui englobe toute la salle de concert : l'entrelacs électronique qui passe d'un « temps » à l'autre.

La perception de plusieurs temporalités simultanées est une chose éminemment délicate à réaliser notamment lorsque les couches temporelles sont denses et se cachent les unes des autres par effet de masquage. Ici, les distances entre instrumentistes sont évidemment salvatrices, mais c'est l'électronique qui produit le liant, qui rend, à ce moment précis, l'intrication possible.



## ORGANISER LE TEMPS : ANTESCOFO ET LES TEMPS MULTIPLES

; NOTE 0 32 Mes217

@proc\_def ::Esolo\_boite217()

```
@abort {
  print "fin de la boite mes117"
  curve percamp @grain := 0.01, @action := ASC0toCS_SYNTN c percbow1.amp ($amperc1*0.12)
  { $amperc1
    { { $amperc1 } @type "cubic_out"
      8 { 0. }
    }
  }
  curve percamp @grain := 0.01, @action := ASC0toCS_SYNTN c percbow2.amp ($amperc2*0.7)
  { $amperc2
    { { $amperc2 } @type "cubic_out"
      8 { 0. }
    }
  }
  ASC0toCS_SYNTN c percbow1.freq (@midi2hz(50))
  ASC0toCS_SYNTN c percbow2.freq (@midi2hz(86))
}
```

Spécification de la sortie de processus

@local \$templink, \$Inst, \$amperc1, \$amperc2

\$amperc1 := 0.  
\$amperc2 := 0.

```
curve perctemp @grain := $grain_curve
{ $templink
  { { 60. } @type "cubic"
    8 { 72. } @type "cubic"
    8 { 48. } @type "cubic"
    9 { 90. }
  }
}
```

Spécification du tempo de la boucle

```
curve percamp_217 @grain := 0.01, @action := ASC0toCS_SYNTN c percbow1.amp ($amperc1*0.12)
{ $amperc1
  { { 0. } @type "cubic"
    4 { 0.16 }
    9 { 0.05 }
    4 { 0.16 }
    10 { 0.05 } @type "cubic_out"
    20 { 0. }
  }
}
```

```
curve percamp2_217 @grain := 0.01, @action := ASC0toCS_SYNTN c percbow2.amp ($amperc2*1.5)
{ $amperc2
  { { 0. } @type "cubic"
    4 { 0.05 }
    9 { 0.2 }
    4 { 0.1 }
    4 { 0.05 }
    6 { 0.2 } @type "cubic_out"
    20 { 0. }
  }
}
```

```
curve freqflux @grain := $grain_curve
{ $freqflux
  { { 50. }
    8 { 50. }
    7 { 52. }
    8 { 50. }
  }
}
```

```
::ASC0toCS_percuBow("percbow1" 3,50,50,4)
::ASC0toCS_percuBow("percbow2" 3,50,86,4)
```

```
loop freqch 1/3 @tempo := $templink @exclusive
{
  ASC0toCS_SYNTN c percbow1.freq (@midi2hz($freqflux))
  ASC0toCS_SYNTN c percbow2.freq (@midi2hz(($freqflux+36)))
  3/12 ASC0toCS_SYNTN c percbow1.freq (@midi2hz(53))
  ASC0toCS_SYNTN c percbow2.freq (@midi2hz(89))
}
```

100 \_ := 0 ; parce que la clause abort n'est pas (encore) réursive...

::Esolo\_boite217()

obj::spat("bowmes217",50,"csound4","vbap",::spat\_traj\_fun,[::lissaj2,9,40,0,4.]])

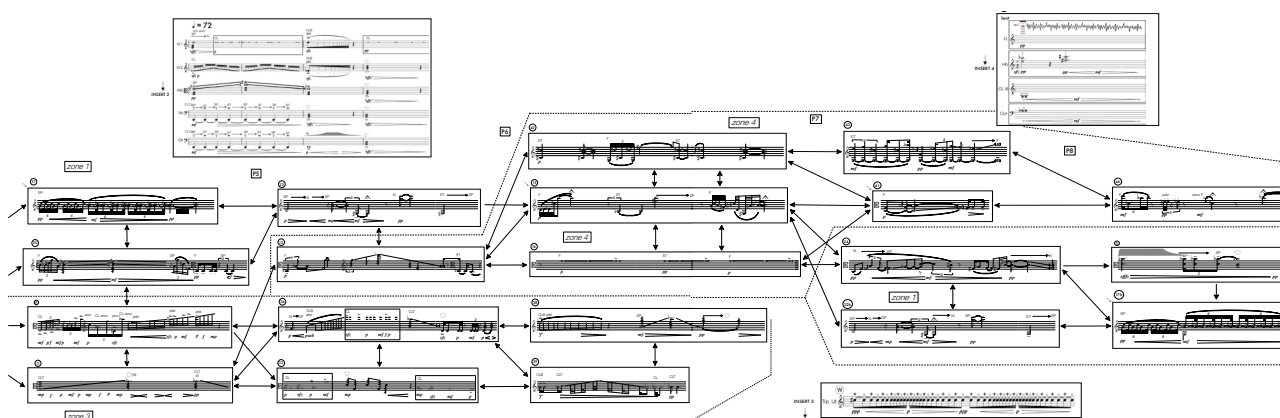
Code Antescofo de la mesure 217 de *Namenlosen*



## Les sentiers qui bifurquent

Dans la partie IV de *Tesla ou l'effet d'étrangeté*<sup>4</sup>, des processus sont définis comme des « zones » qui se déclenchent si l'interprète y « pénètre ». Le soliste a le choix entre divers parcours. Une fois entré dans la zone, les temps des processus liés à cette région sont indépendants de l'interprète. Des actions plus classiques en électronique (déclenchées au fur et à mesure de l'avancement de la partition) servent alors « d'orchestration » du lien entre l'interprète et les processus de région, autant du point de vue temporel que du point de vue du timbre.

Les couches temporelles sont ici indépendantes, l'interprète « suit » le temps de l'électronique, tout en ayant la liberté de passage d'une cellule à l'autre. Pourtant, à certains endroits, l'alto vient « figer » le temps des processus qui cessent, pendant une certaine durée, d'évoluer, autant du point de vue des hauteurs, du timbre que de la spatialisation. L'alto joue alors dans un temps « gelé », jusqu'au moment où il franchit un certain seuil.



Il s'agit ici de composer des rapports dialectiques entre électronique et interprète, jamais unilatéraux mais toujours l'un à l'écoute de l'autre, préservant leur indépendance mais avançant de manière synchrone. La notion d'*électronique de chambre*, notamment conceptualisée par Marco Stroppa, est pour nous fondamentale dans la mesure où le suivi de partition ne doit en aucun cas circonscire l'électronique à une relation unidirectionnelle : de l'instrumentiste vers l'électronique. Ici l'électronique suit l'instrumentiste dans certains modules, mais comporte des éléments qui évoluent sans liens direct avec celui-ci, mis à part lorsqu'il entre dans des zones particulières. L'instrumentiste doit jouer des sortes de fenêtres, dont les temps sont définis de manière assez floue, pour la simple raison qu'ici c'est à lui de suivre l'électronique et de décider, en conséquence, le moment propice pour passer dans une nouvelle région de l'espace musical.

Dans le code ci-contre, les fonctions `whenever` utilisées ont pour fonction de repérer si l'instrumentiste est entré dans une zone particulière et de lancer le processus correspondant à cette zone. Cet exemple montre en particulier l'utilité de l'attribut `@abort` pour les processus (ici un extrait de la définition du processus `::zone1`). Ici, on ne peut savoir à l'avance quand un processus de zone se terminera, il contient un ensemble de boucles qui évoluent dans le temps et qui contrôlent des générateurs de synthèse. Un même générateur peut être utilisé pour 2 zones différentes, ce qui fait que le passage d'une zone à l'autre peut s'avérer extrêmement périlleux, notamment si deux `curves` s'adressent à un même générateur. Ici, les attributs `@abort` permettent de spécifier comment se termine un processus et donc avec celui-ci les éventuelles `curves` qui sont utilisées dans un nouveau processus qui se lancera au même moment.

<sup>4</sup> Pour alto et électronique, commande du Festival Musica 2014.

## ORGANISER LE TEMPS : ANTESCOFO ET LES TEMPS MULTIPLES

```

@proc_def ::Stalker ()
{
; la deuxième condition du whenever sert à éviter de lancer plusieurs fois un même processus
whenever ZONE1 (@member($zone1,$LAST_EVENT_LABEL) && $zone1_status = 0)
{
if ($RIV_status = 0)
{ ::Exils() }
::zone1()
abort ::zone2
abort ::zone4
$zone2_status := 0
$zone4_status := 0
}

whenever ZONE2 (@member($zone2, $LAST_EVENT_LABEL) && $zone2_status = 0)
{
if ($RIV_status = 0)
{ ::Exils() }
print "zone2 reperee"
::zone2()
abort ::zone1
abort ::zone4
$zone1_status := 0
$zone4_status := 0
}

whenever ZONE4 (@member($zone4, $LAST_EVENT_LABEL) && $zone4_status = 0)
{
if ($RIV_status = 0)
{ ::Exils() }
::zone4()
abort ::zone1
abort ::zone2
$zone1_status := 0
$zone2_status := 0
}
}

```

```

@proc_def ::zone1 () @tempo := $tempo_ZONE1
@abort {
SPAT3 aperture 80
SPAT4 aperture 80
lvl_e1 0. 900
lvl_e1tor 0. 900
curve ampg75 @Grain := 0.05s, @Action := ASC0toCS_SYNTL c g7.amp ($kampProcZG7 * $i
{ $kampProcZG7 {
{ $kampProcZG7 }
10 { 0. }
}
}
}
curve ampg85 @Grain := 0.05s, @Action := ASC0toCS_SYNTL c g8.amp ($kampProcZG8 * $i
{ $kampProcZG8 {
{ $kampProcZG8 }
10 { 0. }
}
}
}
curve ampg95 @Grain := 0.05s, @Action := ASC0toCS_SYNTL c g9.amp ($kampProcZG9 * $i
{ $kampProcZG9 {
{ $kampProcZG9 }
10 { 0. }
}
}
}
}
{
$zone1_status := 1
print "-----> process Zone1"

$cptZone1 := $cptZone1 + 1
abort LISSAJOU5
}

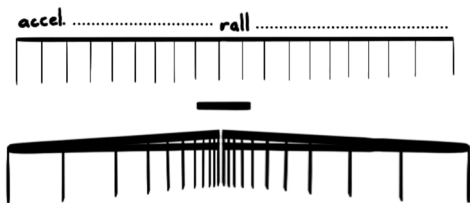
```

Nous avons vu quelques exemples de polyphonies et couches temporelles. Il en existe bien d'autres et l'idée n'était pas d'en faire une liste exhaustive mais plutôt de montrer les potentialités d'écritures de couches temporelles que permet le langage Antescofo. Entre deux couches temporelles, la solution la plus simple est certainement la totale indépendance, dans ce cas les liens qu'il peut y avoir sont sous-entendus dans la partition elle-même (jeu d'orchestration, de hauteurs...) mais ne sont pas directement explicites. On peut *a contrario*, nous l'avons vu, établir des liens directs : deux lignes viennent former une tresse comme dans notre exemple pour violoncelle et électronique ou alors plusieurs lignes temporelles indépendantes sont entrelacées par une ligne qui les « réunit » toutes. On a vu aussi des combinaisons dans lesquelles une partie très indépendante pouvait être altérée par l'incursion d'une autre couche temporelle (celle de l'instrumentiste dans notre exemple). Il y a donc un spectre assez large de rapports possibles, d'intrications entre différentes couches temporelles. Nous discuterons par la suite des questions, nombreuses, qui se posent en terme de notation de ces *multiplicités temporelles*.

## Les fenêtres temporelles

### Le temps « local » du geste

Si on pense le temps de manière plus locale, pour passer à l'échelle du geste, les variations de tempo peuvent être un moyen très expressif d'exprimer un certain phrasé ou d'obtenir des fluctuations temporelles extrêmement fines qui seraient difficiles à exprimer en temps absolu mais aussi en temps relatif. Comme sur une partition « classique », on note alors les événements de manière événementielle, à durée égale, puis on agit directement sur le tempo de ce groupe régulier.



Dans l'exemple de droite, la notation du tempo est très empirique, on spécifie des changements de tempo qui ne font aucun sens (ici variant entre 40 et 130 !) en terme de tempo puisqu'on parle ici plutôt de vitesse d'un ralenti ou d'un accéléré. Il faut généralement tester une valeur, écouter le résultat puis modifier en conséquence.

Si la variation du tempo permet ici de garder une notation rythmique relative et plutôt simple, elle est cependant très peu expressive et oblige le compositeur à écrire des valeurs qui n'ont, pour ainsi dire, que peu de sens musical. Si le compositeur, souhaite ralentir ou accélérer certains passages, il devrait pouvoir le faire sans avoir même à spécifier les tempi « en dur » qui lui corresponde. Nous avons donc réfléchi à un concept de fenêtre temporelle qui finalement nous a mené beaucoup plus loin que les notions classiques de ralenti ou d'accélééré et dont nous allons parler maintenant.

```
GROUP tempfus @tempo := 80
{
  curve tempfusee @grain := 0.1s
  {tempfusee
  {
    4 { 40 } @type "quart"
    1 { 130 } @type "quad"
    1 { 50 }
    3/4 { 80 }
  }
}

GROUP fusee @tempo := $tempfusee
{
  ::SPAT_lissaj2inv("SPAT7",1.4,6,0)
  curve ampexpl0 @grain := 0.05s
  {ampexpl0
  {
    2 { 0.09 } @type "quad"
    1 { 0.15 } @type "quad_in_out"
    1 { 0.04 }
    1/2 { 0.17 }
    1/2 { 0.09 }
  }
}
::ASC0toCS_points("i33",13/6,$ampexpl0,0.9,76)

11/6 ::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,79)
1/6 ::ASC0toCS_points("i33",1/8,$ampexpl0,0.1,78)
::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,78.5)
1/6 ::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,77)
1/6 ::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,80)
::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,80.5)
1/6 ::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,86)
::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,86.5)
1/6 ::ASC0toCS_points("i33",1/10,$ampexpl0,0.7,91)
::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,91.75)
1/8 ::ASC0toCS_points("i33",1/10,$ampexpl0,0.7,90)
1/8 ::ASC0toCS_points("i33",1/10,$ampexpl0,0.7,89)
::ASC0toCS_points("i33",1/8,$ampexpl0,0.7,89.5)
```

## L'expressivité au service du temps

Afin de répondre aux nécessités d'expressivité quant aux spécifications de « modulations temporelles », nous avons pensé à créer une syntaxe symbolique qui définissent les variations de tempo d'un passage. Nous avons de ce fait utilisé des NIMS<sup>5</sup>, qui sont des représentations symboliques de fonctions interpolées, de manière à représenter — de façon beaucoup plus expressive qu'une liste de nombres assez peu explicites — les variations d'un tempo.

Les NIMS linéaires se composent de segments qui montent (+), stagnent (=) ou descendent (-). L'exemple ci-dessous est une courbe linéaire divisée en deux segments correspondant à la figure ci-contre.

```
$nim := NIM {+,-}
_ := @gnuplot($nim)
```

Un symbole correspond à un segment. On peut ajouter un ensemble de symboles : (++,---) donnerait ainsi un segment augmentant de 2 puis descendant de 3. Par défaut un segment a une valeur de durée de 1 et un delta de 1. On peut cependant spécifier une durée et un delta par segment comme dans l'exemple suivant.

```
$nim := NIM {+(1,2),-}
_ := @gnuplot($nim)
```

Cette NIM sera lu par la suite par une courbe qui tiendra lieu de « lecteur de nim », envoyant les valeurs à une variable donnant le tempo d'un groupe ou d'un processus.

Les NIMS polynomiales se composent aussi de segments mais avec des symboles différents : \* monte, \ descent et = stagne. Deux valeurs sont associées à chaque symbole, l'une pour spécifier la tangente au point d'arrivée, l'autre pour la durée. Une 3ème valeur optionnelle indique un delta.

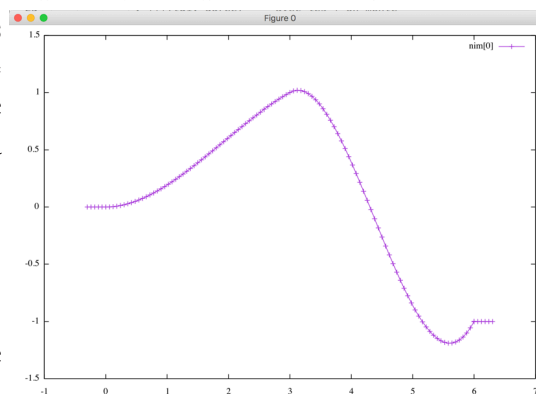
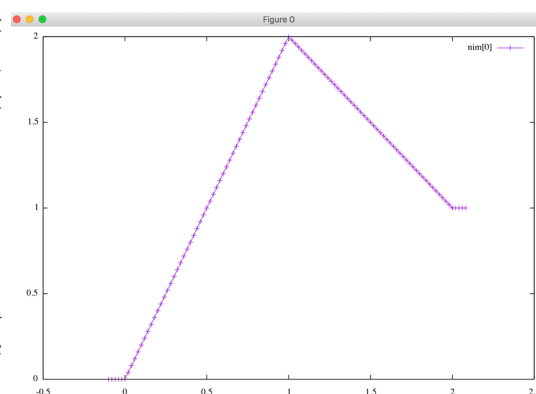
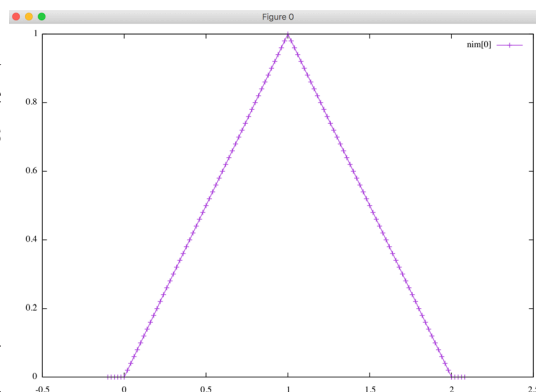
```
$nim2 := NIM {*(0.3,3),\\(1,3)}
_ := @gnuplot($nim2)
```

Les segments linéaires et polynomiaux peuvent être juxtaposés dans une NIM commune. La fonction @gnuplot permet d'afficher la NIM sur un graphique de façon à la visualiser.

On voit clairement ici l'intérêt d'une telle syntaxe si on la couple avec des macros permettant de créer des bibliothèques qui deviennent une sorte de « vocabulaire temporel ». L'exemple le plus simple est celui de l'accélééré-ralenti :

```
@macro_def @acc_rall { +,- }
```

Les segments de la NIM sont signifiés de ce fait qualitativement. Il n'est plus nécessaire d'entrer des valeurs qui ne font pas sens *musicalement*. Nous allons voir à présent comment adapter le tempo d'un groupe, de façon à ce qu'il respecte une de ces NIMS « qualitatives ».



<sup>5</sup> Les nims sont des représentations symboliques de fonctions interpolées.

## Une fenêtre de temps...

La fonction @adapt a été écrite pour répondre à une idée bien précise. Nous souhaitons pouvoir écrire, dans une partition électronique et de la manière la plus expressive, ce qu'on a appelé des *fenêtres temporelles*. L'idée était de pouvoir écrire un groupe, avec des événements placés dans le temps de ce groupe, qui représenterait une voix indépendante mais sur qui on apposerait un tempo ou une évolution de tempo qui serait ensuite recalculée de façon à ce que cette fenêtre viennent se superposer à une voix principale (l'instrumentiste par exemple), entre des points de départ et d'arrivée.

Il s'agit de réaliser ce qui a été relativement souvent tenté en musique instrumentale, avec plus ou moins de succès, et qui consiste à superposer deux voix temporelles synchronisées en début et en fin. Ici, la fonction permet d'écrire des choses un peu plus puissantes. En effet, non seulement le tempo du groupe « fenêtré » est recalculé mais on peut lui imposer une certaine évolution, notamment grâce aux NIMS qualitatives que nous venons de présenter.

Ci-dessous la définition de la fonction @adapt permettant de recalculer le temps des événements d'un groupe selon une variation de tempo et une durée totale :

```
@fun_def @adapt($final_dur, $nim, $actual_dur)
{
    @local $start := $nim.min_key()
    @local $stop := $nim.max_key()
    @local $nim2 := $nim.scale_x($final_dur/($stop - $start))
    @local $total_beat, $nim3

    $total_beat := $nim2.integrate0()[0]
    $nim3 := $nim2.scale_y(60. * $actual_dur/$total_beat)

    return $nim3
}
```

Le paramètre \$nim de la fonction définit le profil « symbolique » de tempo à suivre et la fonction @adapt transforme ce profil de tempo en une fonction (nim) qui adapte ce tempo pour une courbe de tempo suivant le même profil mais permettant d'atteindre la position \$final\_dur (en beat) au bout d'une durée \$actual\_dur (exprimée aussi en beat). La fonction applique une transformation linéaire en  $x$  et en  $y$  sur la nim pour s'adapter à la durée. Le rescaling en  $y$  nécessite le calcul de l'intégrale du tempo pour spécifier la position à atteindre.

Il suffit par la suite de créer une `NIM` qualitative qui décrit la courbe de tempo souhaitée et de la donner comme argument à la fonction `@adapt`, contrôlant le tempo du groupe. Dans l'exemple ci-dessous, c'est une simple boucle itérative qui va donc suivre le tempo décrit par la `NIM`. Le tempo sera réadapté pour que le « temps-réel » du groupe soit égal à la durée donnée par la variable `$durfinal`.

```
$nim := NIM { =, \ (1, 3), = }
$durfinal := 5
$tempo := @adapt($durfinal, $nim, @specified_duration("G"))
GROUP G @tempo := $tempo
{
  loop toto 1/12
  {
    bang bang
  }
  5 abort toto
}
```

La durée « écrite » du groupe est calculée par la fonction `@specified_duration` et la durée imposée totale est donnée comme premier argument de la fonction `@adapt`.

## Stratégies de synchronisation : le temps du phrasé

*« Tout ce qui va arriver peut et doit être prévu.  
Réciproquement, tout ce qui a été prévu doit obligatoirement arriver. »*  
Devise Shadok.

L'interprétation par un musicien d'une partition musicale implique un degré de liberté sur certains paramètres. Le temps est une propriété qui n'est pas totalement déterminée lors de la phase compositionnelle, et qui se trouve en partie soumise à l'interprétation du musicien. Dans le contexte particulier de la musique mixte, pour synchroniser des actions électroniques au jeu d'un musicien il faut à la fois tenir compte :

- du contexte musical (prédéfini),
- du tempo réel du musicien dépendant de l'interprétation de celui-ci,
- de l'occurrence des événements musicaux,
- des erreurs éventuelles du musicien et de la machine d'écoute.

Le langage Antescofo permet de modéliser et d'expliciter l'ensemble de ces caractéristiques à travers la mise à disposition de stratégies de synchronisation et de rattrapage d'erreurs. Selon le *contexte musical*, le compositeur peut ainsi définir un type de synchronisation qui déterminera le comportement des actions électroniques.

Dans un contexte de musique de chambre, les interprètes combinent différentes techniques pour se synchroniser : un tempo partagé, une écoute réciproque et des informations visuelles (gestes, regards). On retrouve dans Antescofo une modélisation des deux premières techniques mais pas de la troisième. En contrepartie, le système possède un temps de réaction bien meilleur que celui du musicien et peut être considéré comme nul d'un point de vue perceptif. Cependant le système doit également anticiper les événements. Il se trouve que la bonne manière d'anticiper dépend du contexte musical. Les problèmes de synchronisation et de coordination musicales ont beaucoup été étudiés dans la littérature<sup>6-7</sup>. L'estimation du tempo dans Antescofo est d'ailleurs le fruit de l'une de ces études<sup>8</sup>.

6 Palmer C. Loehr JD, Large EW. *Temporal Coordination and Adaptation to Rate Change in Music Performance*. Journal of Experimental Psychology : Human Perception and Performance, August 2011.

7 Alan M. Wing, Satoshi Endo, Adrian Bradbury, and Dirk Vorberg. *Optimal Feedback Correction in String Quartet Synchronization*. Journal of The Royal Society Interface, 11(93), 2014.

8 Edward W. Large. *Periodicity, Pattern Formation, and Metric Structure*. Journal of New Music Research, 22 :173–185, 2001.

José Echeveste a créé un ensemble de stratégies de synchronisation qui permettent d'expliquer précisément comment une séquence d'actions se synchronisera, en temps réel, avec les événements du musicien. Nous discuterons dans ce chapitre des différentes stratégies de synchronisation et des contextes musicaux qui privilégient telle ou telle approche. Nous verrons que ces stratégies peuvent apporter une grande souplesse à l'électronique, et qu'elles amènent des possibilités d'écriture qui étaient difficilement envisageables auparavant sans l'utilisation d'un click-track. Elles permettent, nous le verrons, d'approcher parfois la notion de phrasé, si importante en musique et qui était relativement absente du vocabulaire de la musique électronique.

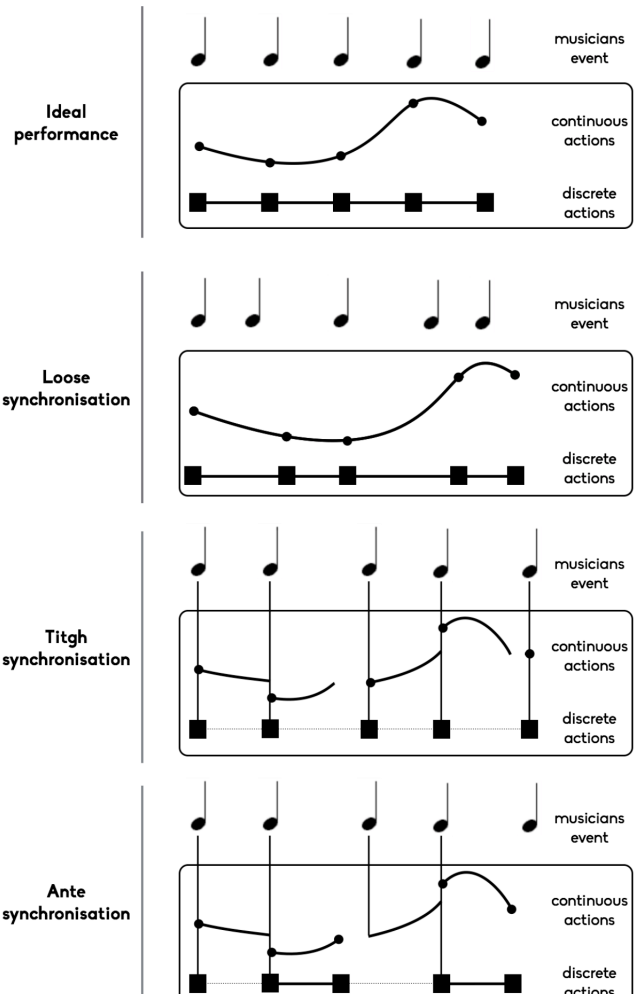
## Stratégies de synchronisation avec tempo non-adaptatif

Lors de l'interprétation, le musicien dévie en permanence des temps « absolus » de la partition. Lorsque Antescofo détecte un événement, une note jouée par l'instrumentiste, il tente alors de rectifier les temps des actions électroniques de façon à ce que les prochains événements soient synchronisés de la meilleure des façons avec l'interprète. Nous verrons que les manières de rectifier les délais des actions d'un groupe varient selon les types de synchronisation et les éléments qui sont pris en compte (tempo, position des actions). De ce fait, chaque type de synchronisation implique un certain degré de continuité ou de discontinuité. Nous donnerons quelques exemples permettant d'appréhender ces stratégies afin de les adapter aux contextes dans lesquels elles sont utilisées.

### Stratégie lisse (loose)

Si l'on ne spécifie pas de stratégie particulière, la synchronisation par défaut d'Antescofo sera de type @loose, ce qui signifie que la vitesse d'écoulement des délais entre les actions d'un groupe ne dépendra que de l'estimation du tempo en temps-réel. La position des actions électroniques ne sera pas mise à jour à la détection d'un nouvel événement ce qui implique que seul le tempo sera susceptible de changer. De ce fait, la position des actions peut se trouver décalée par rapport aux événements de l'instrumentiste (les notes jouées). On peut voir ci-contre un schéma permettant de comprendre les différents comportements des stratégies de synchronisation dont nous parlons dans ce chapitre.

La stratégie de synchronisation @loose permet, si elle est associée à une action continue comme une curve, de préserver sa continuité et d'assurer des évolutions fluides. Si une action continue est lancée



« Comportement des actions continues (courbes) et discrètes (carrées) en fonction de la stratégie de synchronisation choisie. Le premier cas correspond à une interprétation idéale (le musicien joue exactement au tempo de la partition) ; le choix de la stratégie n'a aucune influence. Les trois autres cas correspondent aux trois stratégies pour une même interprétation non idéale : la deuxième et la quatrième notes sont jouées en avance, la troisième et la cinquième en retard. La stratégie loose transforme continuellement une action continue, mais l'alignement des actions et des événements est perdu. La stratégie tight préserve cet alignement, mais introduit des discontinuités dans les actions continues. La stratégie ante aligne les actions avec les événements quand ceux-ci ne sont pas en retard. » (illustration et légende tirées de la thèse de doctorat de José Echeveste).

et que l'instrumentiste joue un peu plus vite, la stratégie @loose permettra donc d'éviter les sauts qui, selon ce qui est contrôlé par la curve (fréquence, amplitude...), peuvent véritablement « sauter aux oreilles ».

### Stratégie événementielle (tight)

La stratégie @tight d'un bloc d'actions permet de maintenir la position des actions au plus près de celles des événements, chaque action étant exécutée après l'action ou l'événement qui la précède.

Si l'on utilise une curve, contrairement à la stratégie @loose, une stratégie @tight pourra provoquer des sauts — de façon à rattraper un retard — ou s'arrêter à une valeur en attendant l'évènement synchronisant. Cette stratégie sera donc plus adaptée à des actions électroniques ponctuelles qu'à des actions continues (voir le schéma correspondant).

Dans l'extrait de code ci-dessous, la stratégie @tight est utilisé pour synchroniser des actions atomiques. Les curves intégrées au groupe ne sont ici pas problématique puisqu'elles contrôlent des paramètres associés à ces actions atomiques qui s'avèrent être des attaques brèves d'électronique. Ainsi, dans ce contexte, si discontinuité il y a, elle ne sera pas entendue et les actions atomiques pourront être au plus près des événements du musicien.

```
GROUP doubl
{
  SPAT7 xy 0 1
  curve SPAT7az @grain := 0.07s, @action := {SPAT7 az $SPAT7az}
  {
    { $SPAT7az
      { 360. }
      2 { 400. }
      3 { 380. }
    }
  }
  curve ampexplo @grain := 0.05s
  { $ampexplo
    { 0.05 } @type "cubic"
    6/2 { 0.06 } @type "cubic_out"
    4/2 { 0.05 }
  }
}

::ASC0toCS_points("i33",1/6,$ampexplo,0.2,80)
1/6 ::ASC0toCS_points("i33",1/6,$ampexplo,0.2,77)
1/6 ::ASC0toCS_points("i33",1/6,$ampexplo,0.2,55)
1/6 ::ASC0toCS_points("i33",1/6,$ampexplo,0.2,77)
1/6 ::ASC0toCS_points("i33",1/6,$ampexplo,0.2,80)
1/6 ::ASC0toCS_points("i33",1/6,$ampexplo,0.2,77)

1/6 ::ASC0toCS_points("i33",1/6,$ampexplo,0.2,91)
1/6 ::ASC0toCS_points("i33",1/6,$ampexplo,0.2,77)
1/6 ::ASC0toCS_points("i11",1/6,$ampexplo,0.2,80)
1/6 ::ASC0toCS_points("i11",1/6,$ampexplo,0.2,77)
1/6 ::ASC0toCS_points("i11",1/6,$ampexplo,0.2,55)
1/6 ::ASC0toCS_points("i11",1/6,$ampexplo,0.2,77)
```

Synchronisation d'un groupe  
d'actions atomiques

### Stratégie anticipante (ante)

En plus des problèmes éventuels de discontinuité, la stratégie @tight peut avoir des effets indésirables de ralentissement dans une situation d'interaction forte entre le musicien et l'électronique. Avec cette stratégie, bien que la machine réagisse quasi instantanément aux événements du musicien, celui-ci ne pourra jamais réagir instantanément aux événements de la machine car elle sera toujours en attente du musicien. Il aura donc tendance à ralentir à cause du manque d'initiative de la part de la machine. Pour remédier à ce phénomène une stratégie anticipative @ante est proposée. Cependant, la discontinuité produite par la stratégie @tight sera relativement similaire avec une stratégie @ante.



*Dans la pratique...*

Il peut s'avérer délicat de choisir un type de synchronisation. Avec un peu de pratique, il devient plus aisé de savoir, avant même d'avoir écouté le résultat dans les conditions de la performance, quelle stratégie correspondra à tel contexte. Cependant, l'utilisation d'une simulation s'avère souvent salvatrice, permettant de tester les différentes stratégies dont les différences, en certaines circonstances, peuvent se montrer subtiles. Dans tous les cas, il sera préférable de rassembler les actions de même type (atomiques, continues) dans des groupes distincts, de façon à attribuer à chacun la stratégie qui lui convient.

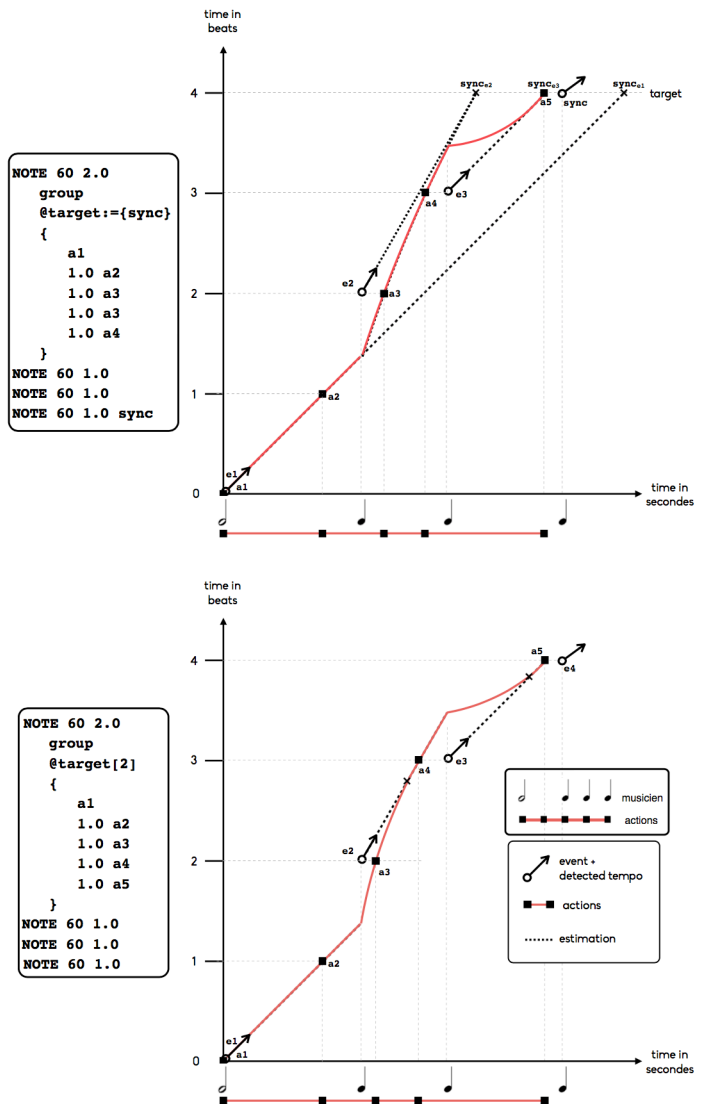
### Stratégies anticipatives d'adaptation du tempo

Les stratégies de synchronisation que nous avons présenté jusque là ne proposent pas de véritable interaction entre le musicien et les actions électroniques. Soit la stratégie ne dépend que du tempo du musicien entraînant parfois un décalage de positions que seul le musicien peut rattraper (stratégie @loose), soit elle est complètement réactive aux événements (stratégies @tight et @ante) créant des discontinuités dans la temporalité du processus. Dans tous les cas, le tempo du bloc utilisant ces stratégies est toujours égal au tempo estimé du musicien.

Les stratégies que nous allons maintenant présenter visent à répondre à ce problème. Elles sont qualifiées de stratégies anticipatives et visent à calculer dynamiquement et localement le tempo d'une séquence pour une synchronisation lisse et sans à-coups, suivant des indications qui permettent une adaptation souple au contexte musical.

*Cibles statiques*

Tous les événements d'une partition ne sont pas équivalents. Certains ont une importance plus grande que d'autres et forment par leurs présences des points saillants à l'intérieur d'une phrase. Si la spécificité de ces événements est sémantique, elle implique également un comportement particulier pour les musiciens. En musique de chambre, ces points « pivots » seront des moments où les instrumentistes se resynchronisent. Il ne s'agit pas seulement des débuts et des fins de phrase. Entre chaque pivot, la synchronisation sera bien plus souple et rendra alors l'interprétation fluide.



Exemple d'exécution de partitions avec des stratégies de synchronisation avec cibles. Les graphiques montrent la relation entre le temps relatif des actions et des événements (en ordonnée) par rapport au temps absolu (en abscisse). Les actions sont représentées par des carrés et le trait continu symbolise la progression des actions. Les événements sont symbolisés par des vecteurs où la pente correspond à l'estimation du tempo. La pente (le tempo) des actions est calculée à chaque réception d'un nouvel événement. Elle correspond à une fonction quadratique qui dépend de la différence entre la position de l'électronique et celle du musicien et de la contrainte de synchronisation (synchronisation en e4 pour la stratégie adaptative statique et fenêtre de synchronisation de 2 pour la stratégie dynamique). (illustration et légende tirées de la thèse de doctorat de José Echeveste).

La stratégie de synchronisation `@target` associée à une liste d'événements pivots indiqués par le compositeur a pour but d'imiter ce comportement dans le séquençage des phrases électroniques. Pendant la performance, le tempo local du bloc concerné par cette stratégie de synchronisation est dynamiquement ajusté pour respecter ces contraintes temporelles (voir le graphique ci-contre).

Lorsque le système réactif d'Antescofo reçoit une nouvelle valeur de position ou de tempo du musicien, le tempo local au bloc est recalculé pour que la position locale du bloc suive une trajectoire parabolique (dans un repère où le temps est porté en abscisse, la position en ordonnées) qui vérifie les trois points suivants :

- la trajectoire passe par le point courant (position dans le bloc à l'instant courant) ;
- la trajectoire passe par l'événement synchronisant (le pivot) ;
- la tangente (tempo) en ce dernier point est égale au tempo du musicien.

Dans l'exemple suivant, le tempo local du groupe est d'abord calculé en fonction de l'estimation de la date d'arrivée de *e5* puis de *e10*. Cette estimation varie au cours du temps au fur et à mesure des informations fournies par la machine d'écoute. Quand l'estimation varie, le tempo local est recalculé.

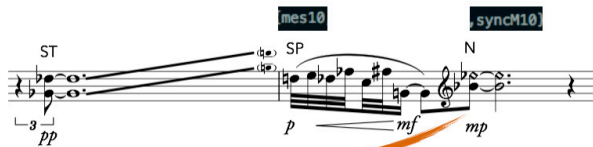
```
NOTE 60 2.0
  group @target:={e5, e10}
  {
    actions...
  }
  events...
NOTE 45 1.2 e5
  events...
NOTE 55 1.2 e10
```

Les attributs `@conservative` et `@progressive` paramètrent le calcul de la position du musicien dans la stratégie de synchronisation. Ils ne sont pertinents que pour les stratégies `@tight` et `@target` où les événements et le tempo sont utilisés pour estimer la position du musicien.

Avec l'attribut `@conservative`, l'occurrence des événements est plus fiable que l'estimation du tempo pour calculer la position du musicien. Ainsi, lorsque la date anticipée d'un événement est atteinte, la position calculée est bloquée jusqu'à ce que cet événement se produise.

Avec l'attribut `@progressive` (la valeur par défaut), l'estimation de la position continuera à avancer même si l'événement prévu n'est pas détecté.

Les stratégies `@target` peuvent donc être combinées à un attribut `@conservative` ou `@progressive` pour permettre d'ajuster la manière dont l'électronique réagira aux fluctuations de l'instrumentiste. Pour donner un simple exemple, si un glissando doit absolument suivre l'instrumentiste et rester exactement parallèle, alors il faudra privilégier l'attribut `@conservative`. Si l'important est davantage de respecter la « ligne » du glissando électronique et d'éviter les sauts ou les « gels », alors on préférera l'attribut `@progressive`.



```
MULTI ((Ab3 Eb4) -> (A4 E5)) 19/3 mes9
GROUP synth @target{mes10,syncM10} @tight
{
  ::ASC0toCS_SYNTH_L("g1",20,0.,0,12,88,0.1,1.2,35,2)
  0.01s curve ampg1 @Grain := 0.08s, @Action := ASC0toCS_SYNTH_L c g1.amp $ampg1
  {
    { Sampg1
      {
        { 0. }
        4 { 0.2 }
        4 { 0.2 }
        1/4 { 0.1 }
        2 { 0.2 }
      }
    }
  }
  curve freqG1 @Grain := 0.08s, @Action := ASC0toCS_SYNTH_L c g1.freq (@midi2hz(
  ))
  { $freqG1
    {
      13/2 { 88. }
      2 { 94. }
    }
  }
  0.01s ::ASC0toCS_SYNTH_L("g2",20,0.,0,7,100,0.1,1.2,35,3)
  curve ampg2 @Grain := 0.08s, @Action := ASC0toCS_SYNTH_L c g2.amp $ampg2
  {
    { Sampg2
      {
        { 0. }
        4 { 0.2 }
        4 { 0.2 }
        1/4 { 0.1 }
        2 { 0.2 }
      }
    }
  }
  curve freqG2 @Grain := 0.08s, @Action := ASC0toCS_SYNTH_L c g2.freq (@midi2hz(
  ))
  { $freqG2
    {
      { 100. }
      13/2 { 100 }
      2 { 94 }
    }
  }
}
```

Il s'agit là d'un exemple simple (tiré de *Tesla ou l'effet d'étrangeté*), avec deux points pivots en début et en fin de phrase. L'électronique commence un glissé de synthèse qui s'achèvera sur la dernière note. Ici le mode @target statique est couplé à une stratégie @tight (sur la dernière version d'Antescofo, le @tight devrait être transformé par @conservative) qui, dans la pratique, s'est avéré la plus adéquat.

### Cibles dynamiques

Il n'est pas toujours utile de définir des points pivots précis. Parfois, il peut être plus profitable de définir un certain niveau de fluence pour un passage donné. Parfois un moment musical peut nécessiter une synchronisation relativement raide. Et parfois une ligne d'électronique doit être la plus souple possible pour accompagner le jeu instrumental d'un passage particulièrement labile.

Dans le cas des cibles dynamiques on ne spécifie plus une liste d'événements synchronisants mais un horizon temporel relatif de synchronisation exprimé en pulsations ou en secondes. À chaque nouvelle valeur de position ou de tempo du musicien, le tempo local est recalculé pour que la position locale du bloc suive une fonction parabolique. La fonction permet une convergence en tempo et en position mais la cible est mouvante et dépend cette fois-ci du paramètre (la fenêtre) donné par l'utilisateur entre crochets. Ce paramètre correspond au temps nécessaire pour converger si la différence entre la position du musicien et la position locale du bloc est égale à 1.

```
NOTE 60 2.0 e1
  group @target := [2s]
  {
    actions...
  }
```

Plus la valeur du paramètre sera grande plus le temps de synchronisation sera long et l'électronique réagira de façon souple. Plus le temps sera court et plus les actions électroniques seront réactives. Le compositeur peut choisir précisément la vitesse de synchronisation ; la valeur 0 étant équivalente à une synchronisation @tight. Il est également possible de varier ce paramètre pendant l'exécution de la séquence car il peut apparaître sous la forme d'une expression qui est évaluée à chaque calcul. La stratégie de synchronisation est alors changée dynamiquement. On peut donc imaginer des processus synchronisés de manières extrêmement raide, avec une fenêtre proche de 0, qui peu à peu deviennent plus souples.

L'exemple ci-dessous est un exemple assez archétypal puisqu'il est constitué d'un rythme quasiment fixe pour la partie instrumentale. C'est un bon exemple dans la mesure où, étant donné que son débit rythmique est assez régulier, mais que l'instrumentiste pourra prendre beaucoup de liberté (il s'agit de surcroît d'un passage difficile instrumentalement parlant et qui nécessite pour l'interprète d'avoir un tempo relativement libre), le choix de la taille de la fenêtre de synchronisation sera tout à fait intéressant à faire varier. Ici, l'électronique, constituée essentiellement de lignes de synthèse va se synchroniser à chaque nouvelle pulsation. C'est une fenêtre relativement serrée et dans le cas où l'instrumentiste bougerait de façon assez importante en terme de tempo, il serait plus judicieux de prendre un horizon temporel de quatre ou 8 pulsations, en particulier sur un passage fait de lignes continues.



```
NOTE D5 1/6
GROUP S3 @target [1] @tempo := $tempoS3
{
  1/2 ::SPAT_lissaj2inv("SPAT3",1.5,18,0)
  ::ASC0toCS_SYNTH_L("g3",250,0.,7,0.55,79,0.,0.2,5,3)
  0.01s curve ampG3 @grain := 0.05s, @action := ASC0toCS_SYNTH_L c g3.amp ($ampG3*$ampGen
  )
  { $ampG3
    { { 0. } @type "cubic"
      1 { 0.2 }
      1 { 0.2 } @type "cubic_out"
      5/2 { 0.05 } @type "cubic"
      1 { 0.15 } @type "cubic_out"
      1/8 { 0.05 }
      23/8 { 0.15 }
      3/2 { 0.2 } @type "cubic_out"
      5/2 { 0.1 }
      3 { 0.15 }
      3 { 0.1 }
      2 { 0. }
      1/2 { 0. } @type "cubic"
      7/2 { 0.2 }
    }
  }
}
```

Synchronisation d'une action continue

On voit donc ici toutes les richesses « interprétatives » que permet ce spectre de synchronisations. On aurait pu penser que dans un contexte où l'électronique serait composée de nappes continues, les subtilités de synchronisations auraient pu être superflues. Hors, il s'agit tout au contraire de composer la manière dont l'électronique va répondre à la liberté temporelle du musicien. C'est au compositeur de choisir de quelle manière l'électronique viendra « entourer » l'interprète instrumental.

## Vers un phrasé électronique ?

Nous l'avons vu, face à cette combinatoire de stratégies, le compositeur doit expérimenter — car c'est une *pratique* — les divers types de combinaisons de synchronisation afin de donner à l'électronique des espaces plus ou moins fluents, combinés à des points de synchronisation pouvant devenir extrêmement précis. Il peut également choisir d'avoir à certains moments une électronique très rigide, et lui donner par cela « l'ascendant » sur l'instrumentiste. Les liens entre deux interprètes (qu'ils soient humains ou électroniques) ne doivent pas être à sens unique.

On a donc ici une position hybride, entre composition et spécifications d'interprétation, dans laquelle le type de synchronisation permet d'approcher un réel travail de musique de chambre, avec les choix compositionnels qu'il nécessite. Si l'on combine la spécification du type de synchronisation et la possibilité de moduler localement le tempo d'un groupe d'actions électroniques, nous pensons qu'il est alors réellement possible d'exprimer un phrasé électronique, avec toutes les fluctuations, accentuations et nuances qu'il sous-entend. Cette question nous semble être un enjeu essentiel, dans la quête d'une écriture pleine et entière de l'électronique. Elle est également fondamentale pour la composition d'une certaine qualité de temps à l'échelle locale.

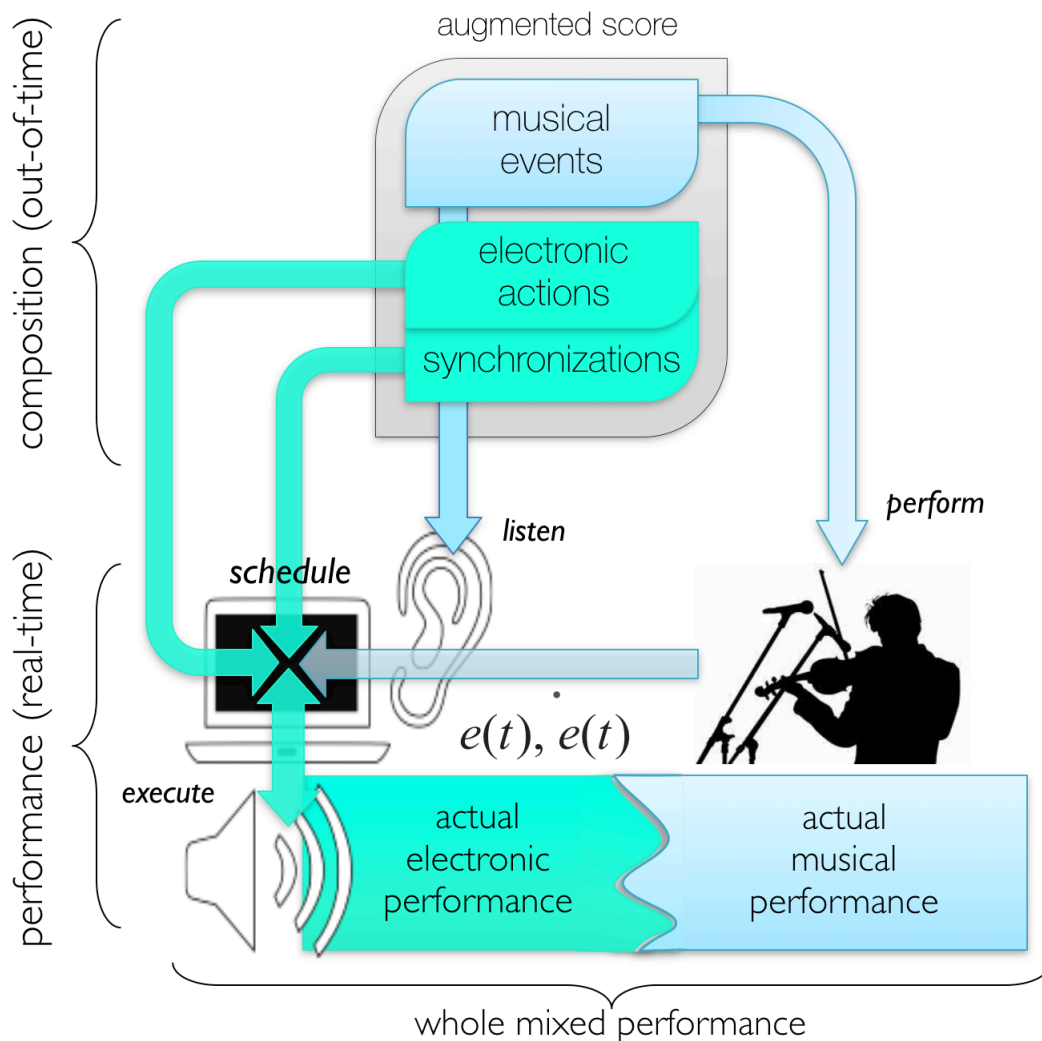


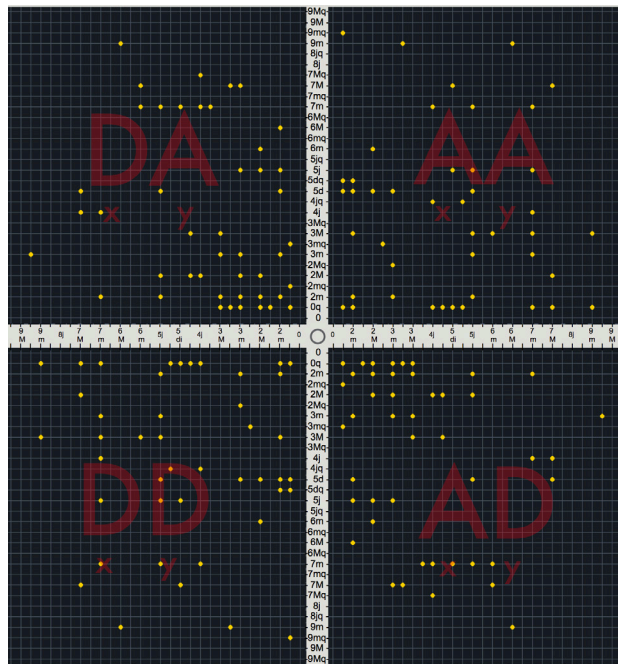
Illustration du système Antescofo réalisée par Jean-Louis Giavitto

## « Mise en temps » d'un système par contraintes

Dans la partie précédente, nous avons évoqué l'utilisation de matrices de successions pour la génération de séquences d'intervalles. Nous présentons ici une implémentation faite en Antescofo pour un module d'aide à la composition mais aussi, dans sa version « temporelle » pour la pièce *Sortir du noir*.

### Matrices de successions

La matrice suivante est traduite sous forme de dictionnaire (la clef correspond à un intervalle et le tableau associé dénote l'ensemble des intervalles suivants possibles) en Antescofo. Rappelons que cette matrice indique les contraintes de successions d'un intervalle à l'autre pour un passage donné. Les intervalles ascendants et descendants sont dissociés.



; matrice d'intervalles Namenlosen

```
$M1 := MAP{ ("ia0q", TAB["ia9mq", "ia5dq", "ia5d", "ia0q", "id0q", "id2mq", "id3mq"]), ("ia2m", TAB["ia6m", "ia5d", "id0q", "id2m", "id2M", "id5j", "id6m"]), ("ia2Mq", TAB["ia3mq"]), ("ia2m", TAB["ia5dq", "ia5d", "ia3M", "ia2m", "ia0q", "id2m", "id3m", "id3M", "id5d", "id5j", "id6M"]), ("ia2mq", TAB["id0q"]), ("ia3M", TAB["id0q", "id2m", "id3m", "id3M"]), ("ia3Mq", TAB["id7m"]), ("ia3m", TAB["ia5d", "ia2Mq", "ia2m", "id0q", "id2m", "id2M", "id3m", "id5j", "id7M"]), ("ia3mq", TAB["ia9m", "id0q", "id7M"]), ("ia4j", TAB["ia7m", "ia4jq", "ia0q", "id2M", "id7m", "id7Mq"]), ("ia4jq", TAB["ia0q", "id2M", "id3M"]), ("ia5d", TAB["ia7M", "ia5j", "ia0q", "id7m"]), ("ia5dq", TAB["ia4jq", "ia0q", "ia5j", TAB["ia7m", "ia5j", "ia5d", "ia3M", "ia3m", "ia2m", "id2m", "id2M", "id5d", "id7m"]), ("ia6M", TAB["ia9m", "id9m"]), ("ia6m", TAB["ia3M", "id7m", "id7M"]), ("ia7M", TAB["ia7M", "ia2M", "ia0q", "id4j", "id5d"]), ("ia7m", TAB["ia7m", "ia5j", "ia4j", "ia3M", "ia3m", "ia0q", "id2m", "id4j"]), ("ia9m", TAB["ia3M", "ia0q"]), ("ia9mq", TAB["id3m"]), ("id0q", TAB["ia3mq", "ia2mq", "ia0q", "id0q", "id5d", "id5dq", "id9mq"]), ("id2M", TAB["ia6m", "ia5j", "ia2M", "ia2m", "ia0q", "id5d", "id6m"]), ("id2Mq", TAB["id3mq"]), ("id2m", TAB["ia6M", "ia5j", "ia5d", "ia3m", "ia2m", "id0q", "id2m", "id3M", "id5d", "id5dq"]), ("id2mq", TAB["ia0q"]), ("id3M", TAB["ia3M", "ia3m", "ia2m", "ia0q"]), ("id3Mq", TAB["ia7m"]), ("id3m", TAB["ia7M", "ia5j", "ia3m", "ia2m", "ia2m", "ia0q", "id2m", "id2Mq", "id5d"]), ("id3mq", TAB["ia7M", "ia0q", "id9m"]), ("id4j", TAB["ia7Mq", "ia7m", "ia2M", "id0q", "id4jq", "id7m"]), ("id4jq", TAB["ia3M", "ia2M", "id0q"]), ("id5d", TAB["ia7m", "id0q", "id5j", "id7M"]), ("id5dq", TAB["id0q", "id4jq"]), ("id5j", TAB["ia7m", "ia5d", "ia2M", "ia2m", "id2m", "id3m", "id3M", "id5d", "id5j", "id7m"]), ("id6M", TAB["ia9m", "id9m"]), ("id6m", TAB["ia7M", "ia7m", "id3M"]), ("id7M", TAB["ia5d", "ia4j", "id0q", "id2M", "id7M"]), ("id7m", TAB["ia4j", "ia2m", "id0q", "id3m", "id3M", "id4j", "id5j", "id7m"]), ("id9m", TAB["id0q", "id3M"]), ("id9mq", TAB["ia3m"]) }
```

Dans une utilisation basique, on pourrait simplement décider d'un premier intervalle puis interroger le système pour savoir quels sont les possibilités pour l'intervalle suivant. Le générateur que nous avons créé permet de générer des listes de hauteurs qui satisfont les conditions de la matrice mais qui répondent aussi à certaines contraintes qui nous sont utiles dans la composition.

## Générateur de séquences

Il s'agit ici de générer des séquences d'intervalles, à partir de la matrice donnée sous forme de dictionnaire, qui satisfassent certaines contraintes comme :

- note de départ et/ou note d'arrivée,
- intervalle de départ,
- direction forcée ou non de la séquence (uniquement ascendante ou descendante)
- ambitus de la séquence ( $\$min$ ,  $\$max$ )

Les fonctions suivantes ont donc été implémentées, la première étant définie ci-après :

- à partir d'une matrice, séquence de  $\$size$  éléments :

```
@HTgenerInt($graph, $startNote, $startInterval, $min, $max, $size)
```

- avec contrainte sur la note finale :

```
@HTgenerInt_Fin($graph, $startNote, $startInterval, $NoteFin, $min, $max, $size)
```

- uniquement ascendant ou descendant :

```
@HTgenerInt_asc($graph, $startNote, $startInterval, $min, $max, $size)
```

```
@HTgenerInt_desc($graph, $startNote, $startInterval, $min, $max, $size)
```

Ce générateur permet notamment de tester une matrice rapidement, d'en percevoir les possibilités et les limites et constitue de ce fait un outil d'aide à la composition particulier. Les contraintes de direction, d'ambitus ou de fin permettent de limiter le nombre de résultats.

```

; -----
; ---- PROCESSUS DE GÉNÉRATION DE SÉQUENCES À n ITÉRATIONS À PARTIR DES MATRICES D'INTERVALLES ----
; -----
; ----- HORS-TEMPS -----
; -----
@fun_def @in_between($min, $note, $interval, $Max)
{
  with $f = $dicoInt($interval)
  in ($min <= $f($note)) && ($f($note) <= $Max)
}
@fun_def @just_asc($note,$interval)
{
  with $f = $dicoInt($interval)
  in ($f($note) > $note)
}
@fun_def @just_desc($note,$interval)
{
  with $f = $dicoInt($interval)
  in ($f($note) < $note)
}

; cherche le premier élément x de $l tel que $f(x) ne soit pas vide
@fun_def @exists($f, $l)
{
  @local $res
  if ($l)
  {
    $res := $f($l[0])
    return (@empty($res) ? @exists($f, @cdr($l)) : $res)
  }
  else { return []}
}

; construit un chemin dans $p
@fun_def @HTgenerInt_($graph, $startNote, $min, $max, $size, $p, $startInterval)
{
  @local $nextNote, $nextIntervals, $new_p
  if ($size == 0) { return $p }
  else
  {
    $nextNote := ($dicoInt($startInterval))($startNote)
    $nextIntervals := [ $x | $x in $graph($startInterval),
                      @in_between($min, $nextNote, $x, $max)]
    $new_p := @concat($p, [$nextNote])
    return @exists(@HTgenerInt_($graph, $nextNote, $min, $max, $size-1, $new_p), @scramble($nextIntervals))
  }
}

@fun_def @HTgenerInt($graph, $startNote, $startInterval, $min, $max, $size) ;-----
{
  @HTgenerInt_($graph, $startNote, $min, $max, $size, TAB[$startNote], $startInterval)
}

```





# Une taxinomie des échelles de temps

« Mon problème, avec les classements, c'est qu'ils ne durent pas ;  
à peine ai-je fini de mettre de l'ordre que cet ordre est déjà caduc.

[...]

Bref, je me débrouille. »<sup>9</sup>

À titre de conclusion de ce chapitre, nous souhaitons ici résumer les différentes échelles de temps avec lesquelles nous pouvons composer dans Antescofo. Chacune d'elles peut être imbriquée ou liée aux autres. Nous avons vu, dans la première partie de cette thèse, l'importance des changements d'échelles et les liaisons — implicites ou explicites — qui peuvent exister entre les différentes couches temporelles d'une œuvre musicale. Il est à noter que différentes échelles de temps peuvent être produites par des moyens identiques. Un processus peut produire un son de synthèse de quelques secondes comme un ensemble complexe s'inscrivant sur une très longue durée.

## Temps des déclenchements

Déclenchement d'action. Calcul de fonction. Assignment de variable. *Simultanéité.*

if - let - @fun - @macro

## Temps des modulations du signal

Variations contrôlées du signal audio. Itérations internes au son. Micro-variations de timbre. Polyrythmies. Focalisation sur un paramètre particulier du son. *Granularité de la durée.*

processus - loop

## Temps des enveloppes

Courbes d'amplitude. Évolutions des paramètres par différents types d'interpolations sur des durées courtes. Lecture de fonctions ou de tables. Suivi de variables externes. *Organisation de la durée.*

nim - curve - setvar

## Temps du phrasé

Stratégies de synchronisation. Fenêtres temporelles. Processus de taille moyenne. Lignes. Groupe d'actions liées sémantiquement. Liaisons de points pivots. Spatialisation. Articulation des actions électroniques. Agogique. *Fluence temporelle.*

group - processus - @target - @tight - @loose - @tempo - curve - nim

## Modulations de tempo (accélééré, ralenti...)

Modulation du tempo d'un groupe ou d'un processus. *Directivité temporelle locale.*

group - processus - @tempo - curve - nim

## Temps des marqueurs formels

Changements radicaux du tempo. Points pivots liés aux synchronicités formelles. *Striage formel.*

whenever - @tempo - \$var

## Temps des processus « globaux »

Processus longs, « d'arrière-fond ». Processus mouvement. *Directivité temporelle globale.*

processus - loop - @tempo

<sup>9</sup> Georges Perec, *Penser/Classer*. Éditions du Seuil, 2003.

# NOTATION ET EXPRESSIVITÉ

« *Sous la métaphore du code, il y a au moins une obsession unificatrice : la dialectique entre loi et créativité, ou — pour reprendre les termes d'Apollinaire — la lutte constante entre l'Ordre et l'Aventure.* »<sup>10</sup>

En 1981, Hugues Dufourt analysait<sup>11</sup> le rôle capital de l'écriture — en tant que *médiation graphique*, projection de la musique et du son sur un espace plan — dans la formation de la pensée musicale. Si l'on veut parler d'écriture pour la musique électronique et la musique mixte en particulier, il faut pour cela pouvoir parler de *notation*, celle-ci jouant un rôle déterminant et structurant dans la composition, plus largement dans la manière de penser et de formaliser la musique. Il n'existe aujourd'hui aucun consensus sur la notation de la musique électronique. On peut raisonnablement se demander si l'existence d'un tel consensus pourrait répondre à l'intégralité des questions posées à la fois par la multiplicité des paramètres et par l'émergence de technologies qui, si elles amènent de nouveaux usages, ne proposent généralement pas de notation musicale correspondante. Le développement de discours théoriques sur la musique électroacoustique est certainement entravé par cet état de fait.

Nous ne proposerons pas ici de nouvelle notation graphique pour la musique électronique. Nous discuterons en revanche des diverses problématiques rencontrées dans l'utilisation d'un langage comme celui d'Antescofo et des difficultés concernant non seulement la notation, mais également les moyens de penser et de composer des processus complexes. S'agissant du langage en lui-même qui se présente sous forme textuelle, nous serons aussi amenés à parler en terme d'expressivité. Le terme d'expressivité est utilisé en informatique pour désigner la capacité d'un langage à exprimer un grand nombre d'algorithmes succinctement et directement. Nous reprenons donc cette notion d'expressivité pour la transposer dans le domaine de la notation et tenter d'estimer si un type de représentation est plus ou moins adapté à l'expression d'une pensée musicale. Il ne s'agit donc pas de trouver un langage universel et commun à tous les compositeurs mais d'intégrer au choix de la représentation graphique et symbolique la question de son expressivité.

Nous montrerons quelques exemples de programmes rendant plus expressive la notation d'éléments musicaux dans la partition Antescofo et discuterons des différents problèmes de représentation de processus complexes. La notation graphique d'Antescofo (AscoGraph) s'est pour le moment figée à un niveau très rudimentaire. Nous présenterons quelques réflexions sur les manques que cela occasionne et sur les possibilités que procureraient une vraie notation ajoutée à la partie textuelle dans Antescofo. Nous étudierons alors trois types de notations différents qui répondent chacun à des utilisations bien particulières et qui nécessitent de ce fait des représentations distinctes : la notation de composition, la notation de réalisation et la notation d'interprétation. Le sujet de cette thèse ne concerne pas directement la notation en elle-même. L'idée ici n'est pas de parler de la notation électronique en général, mais de voir comment la question des temps multiples amènent de nouvelles problématiques de notation et de représentation.

10 Umberto Eco, *Sémiotique et philosophie du langage*. Quadrige / PUF, 1984.

11 Hugues Dufourt, *L'artifice d'écriture dans la musique occidentale*. Critique, 408, 1981.

## La nécessité du symbole

L'utilisation d'une variable dans Antescofo pourrait paraître tout à fait triviale et c'est une pratique absolument classique en informatique. En ce qui nous concerne, c'est pourtant un moyen très expressif « d'éclaircir » une partition Antescofo, en plus d'être un outil indispensable dans la programmation de processus. Les variables globales n'existent par exemple pas dans un logiciel pourtant majoritairement utilisé en musique mixte : Max/MSP (ou sa variante open source Pure Data). Nous allons voir dans ce chapitre un ensemble d'exemples qui mettent à profit, de différentes manières, l'utilisation de variable associée à des processus mais aussi à des bibliothèques propres au compositeur.

Une variable permet d'associer un nom symbolique, porteur de sens, à une donnée, une structure de donnée ou même à un processus mis en route au moment de la déclaration de la variable. Elle peut donc nous amener à signifier ce que représente par exemple le lancement d'un processus au milieu d'une partition. En plus des variables, les processus ou le système d'objet permettent de s'extraire de questions techniques pour se focaliser sur la composition en elle-même lorsqu'il est question d'écrire la partition Antescofo. Loin d'être des gadgets, ils peuvent permettre d'explicitier une pensée musicale et de figurer les liaisons existantes entre deux entités musicales électroniques.

Les anciennes q-lists qui étaient utilisées dans Max étaient très peu expressives. Elles ne permettaient que d'envoyer des messages, qui correspondaient le plus souvent à des listes de paramètres. La « partition » dans ce cas là était une collection de « cues » dont il était peu évident de discriminer les paramètres puisqu'il s'agissait de liste de nombres, de plus ou moins grandes tailles, envoyées à des modules de traitement du signal. L'un des grands apports du langage réactif d'Antescofo est qu'il permet d'utiliser des variables en lieu et place des listes de paramètres qui ne faisaient aucun sens d'un point de vue musical.

### De la création « d'instruments » à la spécification de leurs « modes de jeu »

Lorsqu'on écrit pour un instrument acoustique, sur partition, nous avons sous la main un arsenal de symboles pour faire faire à l'instrumentiste ce qu'on définit comme des « modes de jeu ». Il arrive qu'il soit cependant nécessaire d'utiliser des sortes de tablatures comme notation d'action, lorsque la technique utilisée sort de l'ordinaire et nécessite de ce fait une explicitation du « comment faire ».

Lorsqu'on compose de la musique électronique, on peut voir les modules de synthèse, et même de traitement, comme des instruments. Le problème est que souvent ces instruments doivent être définis « dans l'entièreté de leur être » et donc impliquent un grand nombre de paramètres. L'utilisation classique que nous mettons en œuvre quotidiennement est celle qui consiste à utiliser des macros ou des processus pour lancer des ensembles de paramètres et pour ne pas charger la partition d'informations dont la présence n'a pas d'intérêt en terme de compréhension musicale et sémantique.

L'exemple qui suit est extrait de *Namenlosen* dans lequel l'électronique se compose de modules de synthèse CSound<sup>12</sup> contrôlés par Antescofo. Le code est celui de la deuxième intervention de l'électronique « soliste » qui lance ici une ligne de synthèse CSound. La ligne de synthèse est lancée grâce au processus `::ASCOTOC_SYNTH` puis une source et un type de spatialisation vont lui être affecté via l'objet `obj::spat` sur lequel nous reviendrons par la suite. Seules les informations vraiment pertinentes d'un point de vue compositionnel sont présentes. Ici, les deux *curves* contrôlent l'amplitude et la vitesse d'itération des pics d'amplitude ajoutés à la ligne. Ces courbes sont ici indispensables et donnent une information directe.

<sup>12</sup> Csound est un langage de programmation particulièrement spécialisé pour générer de la musique de synthèse. Ce langage est inspiré de MUSIC, une série de programmes plus anciens développés par Max Mathews.

NOTE C6 1

```

GROUP Esole_mes6 @target [5]
{
  ::ASCtoCS_SYNTH("g2",9,84,0.,7,0.5,0.3,0.01,7,2)
  obj::spat("g2mes6",10,"csound2","vbap",::spat_traj_ade,[[{stp_s,$hb_s,"quad",1},{hb_s,$fl_s,"linear",1},
  ],[$fl_s,$alt_s,"linear",3/2]])

  curve ampG2 @grain := $grain_curve, @action := ASCtoCS_SYNTH c g2.amp $ampG2
  { $ampG2
    { { 0. }
      1 { 0.31 }
      1 { 0.33 }
      1/2 { 0.31 }
      1/2 { 0.31 } @type "quad_out"
      4 { 0. }
    }
  }

  curve iterG2 @grain := $grain_curve, @action := ASCtoCS_SYNTH c g2.tempiter $titerG2
  { $titerG2
    { { 5. }
      1 { 5. }
      1/2 { 8. }
      1 { 12. }
      1 { 8. }
    }
  }
}
}

```

Nous montrons ci-dessous la définition du processus `::ASCtoCS_SYNTH` qui nous permettra de montrer l'ensemble des éléments qui sont en réalité lancés et envoyés par Antescofo à CSound.

```

@proc_def ::ASCtoCS_SYNTH($numGen,$dur,$noteMidil,$kamp,$sicut,$enviter,$iter,$del,$mfo,$numSortie)
{
  // Enveloppe d'itération
  if ($numGen == "g1") { $numTab := 11 }
  if ($numGen == "g2") { $numTab := 12 }
  if ($numGen == "g3") { $numTab := 13 }
  if ($numGen == "g4") { $numTab := 14 }
  if ($numGen == "g5") { $numTab := 15 }
  if ($numGen == "g6") { $numTab := 16 }
  if ($numGen == "g7") { $numTab := 17 }
  if ($numGen == "g8") { $numTab := 18 }
  if ($numGen == "g9") { $numTab := 19 }
  if ($numGen == "g10") { $numTab := 30 }
  if ($numGen == "g11") { $numTab := 31 }
  if ($numGen == "g12") { $numTab := 32 }
  $ampEnviter := $enviter
  $EnvIter := NIM{ 0 0.05, 15 $ampEnviter "exp", 16 0.05 "exp_out"}
  $echNIMenv := [ $EnvIter($enum) | $enum in 0 .. 32 : 1 ] ; echantillonnage de la NIM pour
  envoi de la table

  forall $i in @iota(@size($echNIMenv)) ; ENVOI de la table d'enveloppe (size = 512)
  {
    ASCtoCS_SYNTH wsidx $numTab $i ($echNIMenv[$i])
  }

  ASCtoCS_SYNTH c ($numGen+".AddTurnoff") 0
  ASCtoCS_SYNTH c ($numGen+".freq") (@midi2hz($noteMidil))
  ASCtoCS_SYNTH c ($numGen+".tempiter") $iter
  ASCtoCS_SYNTH c ($numGen+".multidel") $del
  ASCtoCS_SYNTH c ($numGen+".mfo") $mfo
  ASCtoCS_SYNTH c ($numGen+".amp") $kamp
  ASCtoCS_SYNTH e i "\"AddSynth\""" 0 (@b2sec($dur)) ("\" + $numGen + "\"") $sicut $numSortie
  $numTab
}

```

L'ensemble de ces envois, si il est indispensable, ne fait, en terme de notation, que très peu de sens musicalement. Sans l'utilisation du processus, il serait nécessaire d'ajouter à chaque envoi d'une ligne de synthèse, l'ensemble de ces éléments. Cela n'alourdirait pas seulement la partition Antescofo mais amènerait de surcroît à mélanger des éléments « porteurs de sens » et d'autres purement techniques.

Nous voyons donc qu'il est possible, grâce aux objets et processus, de créer des sortes d'instruments, de façon à s'adresser à eux directement et à ne pas surcharger la notation d'éléments non significatifs. Il est souvent fondamental, à un certain stade de la composition, de pouvoir se délester de questions purement techniques. Dans notre exemple, lorsque le processus est créé nous n'avons besoin que d'écrire les informations qui seront véritablement significatives pour la définition de « l'identité » du son.

Dans l'exemple ci-dessous, il s'agit de réaliser des sortes de bibliothèques de « modes de jeu » d'un

instrument de synthèse, propre au compositeur, et ainsi pouvoir écrire comme dans une partition classique des notations symboliques comme *sul pont.* sans avoir à chaque fois à en expliciter les paramètres. On peut surtout passer d'un mode de jeu à l'autre, par interpolation.

```

; --- pour le module Csound utilisant l'opcode "wgbow"
; Declaration des "modes de jeu" pour interpolation
@global $SulPont,$0rd,$SulTasto,$SulPontVib1,$SulPontVib2,$SulTastoVib1,$SulTastoVib2,$0rdVib1,
$0rdVib2

; pression : 0.2 - 5
; position : 0.028 - 0.22

$SulPont := [3.20, 0.032, 3.6, 0.] ; pression, position, freqVib, vibamp
$SulPontVib1 := [3.20, 0.032, 1.3, 0.002]
$SulPontVib2 := [3.20, 0.032, 7.1, 0.003]
$0rd := [2.5, 0.12, 1.3, 0.]
$0rdVib1 := [2.5, 0.12, 1.3, 0.002]
$0rdVib2 := [2.5, 0.12, 7.1, 0.003]
$SulTasto := [4.23,0.2, 1.3, 0. ]
$SulTastoVib1 := [0.9,0.22, 1.3, 0.002]
$SulTastoVib2 := [0.9,0.22, 7.1, 0.003]

; Numero du GenBow, [ModeJeu1,dur1,type1,ModeJeu2, ... , durn,typen,ModeJeu], Grain
; ex ::MdJ_bow("bow1",[$SulPont,1,"cubic",$0rdVib2,2,"linear",$SulTasto] ,0.2)
@proc_def ::MdJ_bow($numBow,$TabMdJ,$grain)
{
  if (@size($TabMdJ)>=4)
  {
    ::M_dj_bow1($numBow,$TabMdJ[0],$TabMdJ[3],$TabMdJ[1],$TabMdJ[2],$grain)
    ($TabMdJ[1]) ::MdJ_bow($numBow,(@cdr(@cdr(@cdr($TabMdJ)))),$grain)
  }
}

@proc_def ::M_dj_bow1($numBow,$MdJ1,$MdJ2,$durInterp,$type,$grain)
{
  @local $interpolMdJB, $x
  ; creation de la courbe entre les valeurs d'interpolations de chaque valeur des 2 tableaux
  $interpolMdJB := [ NIM{ 0 ($MdJ1[$i]), 1 ($MdJ2[$i]) EXPR{ $type } } | $i in @size($MdJ1) ]
  ; "tete de lecture" de la courbe
  curve interp @grain := $grain s,
  @action := { ASC0toCS_SYNTH c ($numBow+".kpres") ([ $nim($x) | $nim in
  $interpolMdJB][0] ) ; prend la 1ere valeur du tableau...
  ASC0toCS_SYNTH c ($numBow+".kpos") ([ $nim($x) | $nim in
  $interpolMdJB][1] )
  ASC0toCS_SYNTH c ($numBow+".kvibf") ([ $nim($x) | $nim in
  $interpolMdJB][2] )
  ASC0toCS_SYNTH c ($numBow+".kvibamp") ([ $nim($x) | $nim in
  $interpolMdJB][3] )}
  { $x
  {
    { 0. }
    $durInterp { 1. }
  }
}
}

```

L'exemple est ici très simple puisqu'il est analogue à l'utilisation « classique » instrumentale. Le module contrôlé par Antescofo étant un modèle physique de corde, la notion de mode de jeu y est tout à fait naturelle. Le code-ci-dessous montre son utilisation dans la partition, avec une interpolation permettant de passer d'un jeu *sul tasto* à un jeu *sul ponticello*.

```

NOTE A#3 1/3

GROUP Bow2
{
  ::ASC0toCS_bow("bow2",16,0.,82,6)
  ::MdJ_bow("bow2",[$SulTasto,5,"quad",$SulPont],0.1)

  curve kampBow2 @grain := 0.05s, @action := ASC0toCS_SYNTH_L c bow2.kampBow $kampBow2
  {$kampBow2
  {
    { 0. } @type "quad"
    1 { 0.2 }
    2 { 0.4 }
    4 { 0.5 }
    8 { 0. }
  }
}
}

```

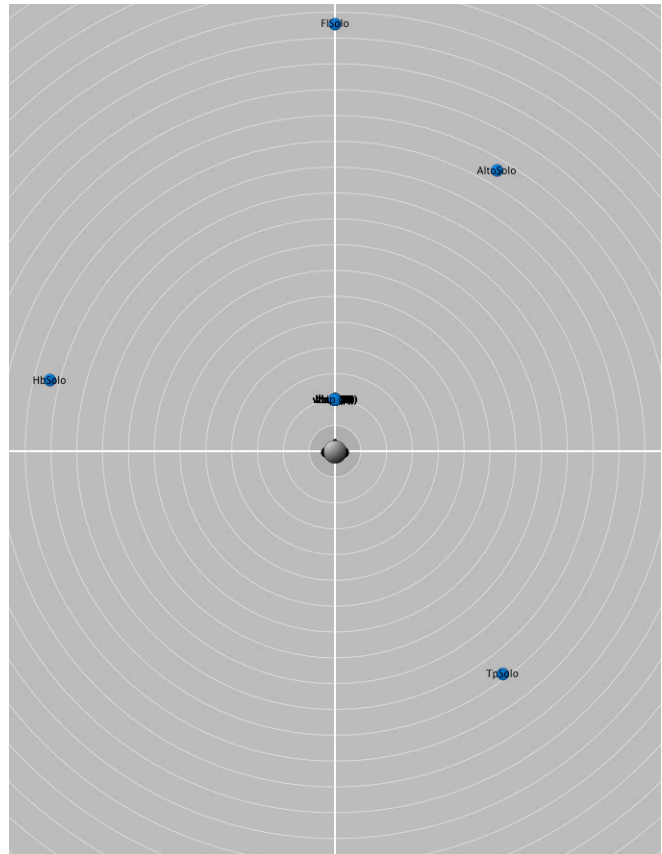
Cette notion de mode de jeu peut être étendue à n'importe quel type de synthèse. On peut imaginer des bibliothèques de configurations associées à des variables symboliques faisant véritablement sens pour le compositeur. Lorsqu'il compose, il pourra alors se focaliser sur des symboles musicalement significatifs plutôt que sur des paramètres.

## Une bibliothèque pour la spatialisation

Lorsqu'on utilise des types de spatialisation complexes (HOA, WFS, totems acoustiques et autres installations non conventionnelles), il convient de créer des moyens de simplifier la notation, dans la même optique que précédemment. Pour la composition de la pièce *Namenlosen*, nous avons créé un objet qui puisse être utilisé dans divers contextes et qui facilite la notation de façon, encore une fois, à focaliser notre attention sur les éléments essentiels à la composition.

*Namenlosen* est une pièce pour 4 solistes, grand ensemble et électronique. Les 4 solistes étant spatialisés, nous souhaitons pouvoir créer des trajectoires de synthèse allant d'un soliste à l'autre. Le code ci-dessous définit les variables associées aux positions des solistes, de façon à spécifier ensuite dans la partition directement : « \$fl\_s, \$alt\_s, 4s » pour signifier un passage de la position du flûtiste à la position de l'altiste en 4 secondes. La variable \$factDist permet d'ajuster l'ensemble des distances de spatialisation (positions et trajectoires) ce qui s'avère très pratique quand on passe du studio à la salle de concert et qu'il faut ajuster l'intégralité des positions et trajectoires sans devoir les changer une par une.

Une stratégie de synchronisation (@target dynamique) est associée à l'ensemble des trajectoires de spatialisation via la variable \$target\_spat\_gen ce qui permet de la changer dynamiquement lorsque cela est nécessaire (selon la nature des événements suivis). Un dictionnaire est créé avec uniquement deux entrées qui représentent les 2 types de spatialisation utilisés. Un tableau est associé à chaque spatialisation (hoa et vbap) et se compose d'un nombre d'éléments qui est défini par les variables \$nombre\_de\_sources\_hoa et \$nombre\_de\_sources\_vbap. Ces tableaux serviront par la suite de tableaux d'assignation des sources de façon à savoir quelle source est disponible au moment où une action électronique doit être spatialisée.



```
; Positions des solistes (ade)
$fl_s := [0.,16.5,8.]
$alt_s := [30.,12.65,7.]
$hb_s := [-76., 12.11,20.]
$tp_s := [143.,11.5,20.]

$Xfl_s := [0.,16.5,2.3]
$Xalt_s := [6.3,10.9,1.5]
$Xhb_s := [-11,2.8,4.1]
$Xtp_s := [6.5,-8.6,3.9]

$factDist := 1.

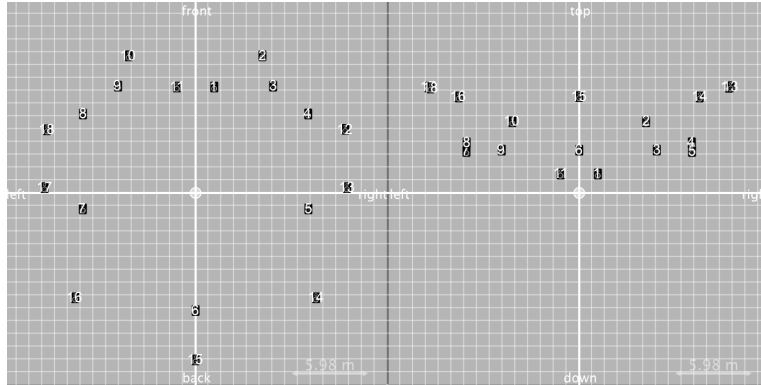
$target_spat_gen := 2 ; stratégie de synchro pour la spatialisation (horizon temporel)
$grain_spat := 0.08 ; grain des courbes d'envoi au spat (en secondes)

; tableaux d'affectation des sources automatique

$nombre_de_sources_hoa := 16
$nombre_de_sources_vbap := 16

; -----
$type_of_spat := map {"vbap",[ 0 | ($nombre_de_sources_hoa)], ("hoa",[ 0 | ($nombre_de_sources_hoa)])}
$assign_spat := map { } ; de type map { ("sound_name", ["Csound1","hoa",1]) }
```

*Namenlosen* utilise deux types de spatialisation différents : HOA pour l'ambisonic et VBAP (voir la disposition des haut-parleurs à la Cité de la Musique ci-dessous). Le nombre de sources nécessaires simultanément est relativement important. Habituellement, lorsqu'on écrit une action électronique, on doit choisir à quelle source elle sera affectée puis choisir une position dans l'espace ou un mouvement particulier. Dans notre cas, nous avons souhaité ne pas nous préoccuper des sources. En effet, le numéro de la source n'a aucun intérêt d'un point de vue musical, contrairement à la position du son dans l'espace, au type de spatialisation (qui ici affecte notamment la réverbération) et aux mouvements.



Le code suivant correspond à la première action de la pièce : une ligne de synthèse partant de la position du hautboïste pour aller vers l'alto puis revenant à la position initiale.

```
::ASCtoCS_SYNTH("g1",8,84,0.,5,0.6,0.3,0.01,7,1)
obj::spat("g1mes4",8,"csound1","vbap",::spat_traj_ade,[[ $hb_s,$alt_s,"linear",2],[ $alt_s,$hb_s,"linear",2]])
```

L'objet `obj::spat` est défini en page suivante. Il permet d'associer un type de spatialisation (ici `vbap`) et une source automatiquement. Un processus de spatialisation est lancé ensuite en parallèle à partir des informations générées par l'objet `spat` et envoie la position de la source au spatialisateur dans Max.

Ci-dessous, le processus utilisé dans le code ci-dessus pour générer la courbe de spatialisation (ici une trajectoire simple) :

```
@proc_def ::spat_traj_ade($sound_name,$stab)
{
    @local $stab_args,$dur_segment

    $stab_args := $stab[0]
    $dur_segment := $stab_args[3]
    if (@size($stab) == 1) { ::spat_traj_ade_($sound_name,$stab_args) }
    else {
        ::spat_traj_ade_($sound_name,$stab_args)
        $dur_segment ::spat_traj_ade($sound_name,@cdr($stab))
    }
}

; -- ::spat_traj_ade_($sound_name,[[ $a1,$d1,$e1],[ $a2,$d2,$e2],$interp,$dur)
@proc_def ::spat_traj_ade_($sound_name,$stab_args)
{
    @local $source,$spat_type,$aa,$dd,$ee,$a1,$a2,$d1,$d2,$e1,$e2,$interp,$dur_segment

    ; arguments du proc
    $a1 := $stab_args[0,0]
    $d1 := $stab_args[0,1]
    $e1 := $stab_args[0,2]
    $a2 := $stab_args[1,0]
    $d2 := $stab_args[1,1]
    $e2 := $stab_args[1,2]
    $interp := $stab_args[2]
    $dur_segment := $stab_args[3]

    $source := (($assign_spat($sound_name))[2])
    $spat_type := ("to_" + (($assign_spat($sound_name))[1]))
    curve traj @grain := $grain_spat s, @Action := @command($spat_type) source $source ade $aa ($dd*$factDist) $ee
    { $aa, $dd, $ee
        {
            $dur_segment { $a1 $d1 $e1 } @type $interp
            $dur_segment { $a2 $d2 $e2 }
        }
    }
}
```

## ORGANISER LE TEMPS : NOTATION ET EXPRESSIVITÉ

```

@obj_def spat($sound_name,$duration,$source_of_sound_name,$spat_type,$traj_type,$procArgs)
{
    @local $obj_temp,$sync_strat,$temp_orig,$source_num

    @init { ; assignation d'une source
        $temp_orig := true
        $sync_strat := $target_spat_gen
        $obj_temp := $RT_TEMPO

        if (@size($type_of_spat($spat_type)) == @reduce(@+,$type_of_spat($spat_type)))
        {
            print "all sources of " $spat_type " are used !"
            print "the " $sound_name "'s spatialization have been aborted"
            abort $THISOBJ
        }
        else
        {
            ::spat_assign($sound_name,$source_of_sound_name,$spat_type)
            print ($type_of_spat($spat_type)) $spat_type $sound_name

            $source_num := (($assign_spat($sound_name))[2])
            group spat_@tempo := ( $temp_orig ? $RT_TEMPO : $obj_temp) @target [
                $sync_strat
            ]
            {
                :: $traj_type($sound_name,$procArgs)

                ; abort final
                $duration let ($type_of_spat($spat_type)[$source_num-1]) := 0
                to_matrix ($matrix_in($source_of_sound_name)) ($matrix_out(($spat_type
                    + $source_num))) 0
                $assign_spat := @remove($assign_spat,$sound_name)
                abort $THISOBJ ; ceci est un suicide !
            }
        }
    }

    @proc_def current_sound_assign()
    { print "chemin de " $sound_name " = " ($assign_spat($sound_name)[0]) " > " ($assign_spat(
        $sound_name)[1]) ($assign_spat($sound_name)[2]) }

    @method_def set_tempo($x)
    {
        $obj_temp := $x
        $temp_orig := false ; pour couper l arrivee de RT_TEMPO pour group spat
    }

    @method_def set_synchro($y) { $sync_strat := $y }

    @broadcast @set_all_spat_tempo($z) { $obj_temp := $x }
}

@macro_def @current_assign_tab($spat_type)
{ print "tableau d'assignation du " $spat_type " [ " ($type_of_spat($spat_type)) " ]" } ; état du
tableau d'assignations

; ----- Affectation d'une source sonore à une source du spat
; - cela évite de devoir se demander à quelle source on envoie le son
; - Ici la fonction ::spat_assign regarde quelles sont les sources libres et assigne
; - automatiquement une source du Spat au module sonore utilisé, puis envoie les infos
; nécessaires à la matrice

@fun_def @fun_assign_spat($tab,$index)
{
    if ($tab[0] == 0) { return ($index+1) }
    else
        { return @fun_assign_spat((@cdr($tab)),($index+1)) }
}

@proc_def ::spat_assign($sound_name,$source_of_sound_name,$spat_type)
{
    @local $source_num
    $source_num := (@fun_assign_spat(($type_of_spat($spat_type)),0))
    $assign_spat := @add_pair($assign_spat,$sound_name,[$source_of_sound_name,$spat_type,$source_num])

    to_matrix ($matrix_in($source_of_sound_name)) ($matrix_out(($spat_type + $source_num))) 1
    let ($type_of_spat($spat_type)[$source_num-1]) := 1
} ; on ajoute une entrée au dictionnaire des assignations
; puis on envoie à la matrice les données correspondant à la clef de dico de matrice
; [CS_synth1,hoa_s1] si la source 1 est libre.
; on met enfin à jour $vbap ou $hoa
; ::spat_assign("synth1mes34","CS_synth1","hoa")

```



On voit à travers cet exemple les possibilités offertes par les diverses combinaisons d'objets et de processus et l'expressivité qu'amène l'utilisation des variables. Il nous semble primordial de pouvoir travailler avec des langages qui permettent de « parler » en termes symboliques, de façon à ne jamais perdre le lien avec nos problématiques musicales. Cela montre également la force d'un langage qui donne à l'utilisateur non pas un outil préfabriqué, mais un arsenal de moyens lui permettant de construire les outils qui seront le plus en adéquation avec sa pensée musicale et avec ses besoins. Ici, une fois les processus et objets définis, il suffit que le compositeur appelle l'objet `obj::spat` pour accéder à l'intégralité des processus de spatialisation.

### « Je déballe ma bibliothèque »

Le fait de pouvoir attribuer un nom à une liste de paramètres, action qui peut paraître assez triviale, nous permet de composer en termes plus qualitatifs que quantitatifs. Cela permet de construire des bibliothèques qui soient en elles-mêmes un matériau *signifiant*. Au fur et à mesure du travail et des pièces, le compositeur se constitue une sorte de bagage compositionnel qui s'agrandit au fil du temps et contribue à singulariser sa pratique.

L'utilisation de bibliothèques et de variables implique d'ajouter un degré symbolique déterminant à la composition de musique mixte. Il peut s'agir, comme nous l'avons vu, de bibliothèques de « modes de jeu » qui permettent de parler symboliquement du timbre et peuvent être adressées à un « instrument » de synthèse, de bibliothèques de trajectoires, de fenêtres temporelles ou encore de bibliothèques de matrices d'intervalles, de suite d'accords ou de spectres. Tout cela constitue un vocabulaire propre au compositeur avec lequel il peut écrire. Plus encore, cela permet de penser de façon multidimensionnelle : une variable représentant un ensemble de paramètres liés à la constitution d'un timbre peut elle-même être utilisée dans un ensemble de plus haut-niveau.

Nous pouvons ici faire un parallèle avec nos espaces compositionnels dont on pourrait par ailleurs implémenter la structure par des structures de données imbriquées dans Antescofo. L'aspect symbolique y est fondamental notamment parce qu'il permet une *appropriation* par l'association, propre à chaque compositeur, qui se joue entre signifiant et signifié.

## Quelques réflexions sur différentes notations

Les éléments dont nous avons parlé jusqu'ici engagent un besoin de représentation. Ils mettent en évidence la prolifération des paramètres à exprimer et impliquent à notre avis une réappréciation de ce que doit être aujourd'hui une partition de musique mixte. La multiplication des paramètres implique à la fois une notation précise à l'intérieur des outils qui les emploient et le choix d'une représentation synthétique à même de résumer l'essentiel de l'identité musicale impliquée par l'utilisation de ces paramètres dans la partition. Il faut donc se poser la question des manières d'organiser, de noter un matériau qui demande autant de spécifications. La question de l'organisation du matériau rencontre ici celle de la notation.

Celle-ci relève alors du choix compositionnel (telle prise de décision musicale étant susceptible d'impliquer telle ou telle notation) et non plus seulement d'un support représentatif neutre. Pour cela, la question d'une notation « universelle » paraît être encore mal posée. Aujourd'hui la notation de musique mixte est un système d'interprétation non uniquement destiné à un instrumentiste. La question de l'interprétation d'une partition implique alors de poser la question de sa représentation : que représente-t-on et pour quel « interprète » (humain, non humain) ?

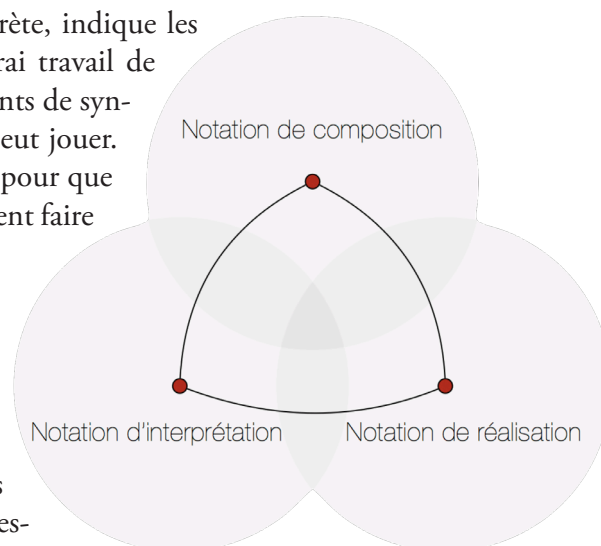
Les nécessités de représentation diffèrent, selon que l'on se place du point de vue de la composition, de la réalisation ou de l'interprétation. Il faut alors adapter les représentations en conséquence et nous pensons qu'il serait donc judicieux de dissocier les types de partitions : partition de composition, de réalisation et d'interprétation. Une partition ne réfère pas forcément ici à une notation exclusivement graphique. Il peut également s'agir d'un programme. Notons qu'il est d'usage de parler de « partition Antescofo », alors même qu'il ne s'agit que d'un langage textuel.

La **partition de composition** est le lieu de la pensée et de l'élaboration musicale dans lequel le degré d'expressivité de la notation est le plus déterminant. La notation se doit ici d'être à la fois un médium de pensée musicale, de fabrication conceptuelle et à la fois un lieu d'expérimentation. Elle doit être en mesure d'exposer en priorité « le sens » plutôt que d'explicitier la manière de produire.

La **partition de réalisation** est quant à elle la plus précise, contenant toutes les informations nécessaires à l'élaboration de l'œuvre musicale. Cette partition est la plus exposée au problème de la prolifération des paramètres. La partition peut rapidement devenir illisible selon le degré d'hétérogénéité des éléments électroniques utilisés. La partition électronique finale peut avoir cela d'étrange qu'elle ne représente pas forcément un résultat sonore mais plutôt la manière de le produire. On trouve cependant ce type de démarche dans le monde purement instrumental, chez Lachenmann notamment.

La **partition d'interprétation**, destinée à l'interprète, indique les éléments les plus essentiels et permet de ce fait un vrai travail de musique de chambre, lui indiquant par exemple les points de synchronisation et les zones plus floues avec lesquelles il peut jouer. Les repères visuels doivent être suffisamment explicites pour que l'interprète puisse les associer aux sons auxquels ils peuvent faire référence.

Dans le cas d'une partition de composition, les notations graphiques et symboliques seront souvent efficaces. Mais certaines représentations, en particulier celles « d'hétérogénéités temporelles » s'avèrent complexes à mettre en œuvre et sont encore aujourd'hui très peu développées. Nous pensons en particulier aux proces-



sus complexes, qui agissent sur des échelles de temps diverses, qui s'exécutent selon certaines conditions extérieures ou dont le temps dépend d'un calcul particulier. La vision linéaire de la timeline classique ne permet pas d'appréhender l'ensemble des processus avec lesquels nous travaillons. Si l'interface textuelle peut être très puissante, en ce qu'elle n'impose pas de représentation normée, elle reste en même temps limitée — justement parce qu'elle ne propose pas d'autres visions que celui de la ligne de code.

Si nous distinguons partition de composition et partition de réalisation, c'est que certains aspects purement techniques (par exemple la courbe des  $x$  et des  $y$  des positions d'une source de spatialisation) n'ont pas d'intérêt à être représentés, d'un point de vue musical, si ce n'est pour encombrer les représentations graphiques inutilement. En revanche ils pourront être utiles lorsqu'il faudra par exemple s'assurer de la bonne juxtaposition de différentes courbes ou tout simplement pour vérifier que « tout est à sa place ». Il est important qu'une notation destinée à la *composition* puisse s'abstraire des aspects qui ne concernent pas directement le propos musical.

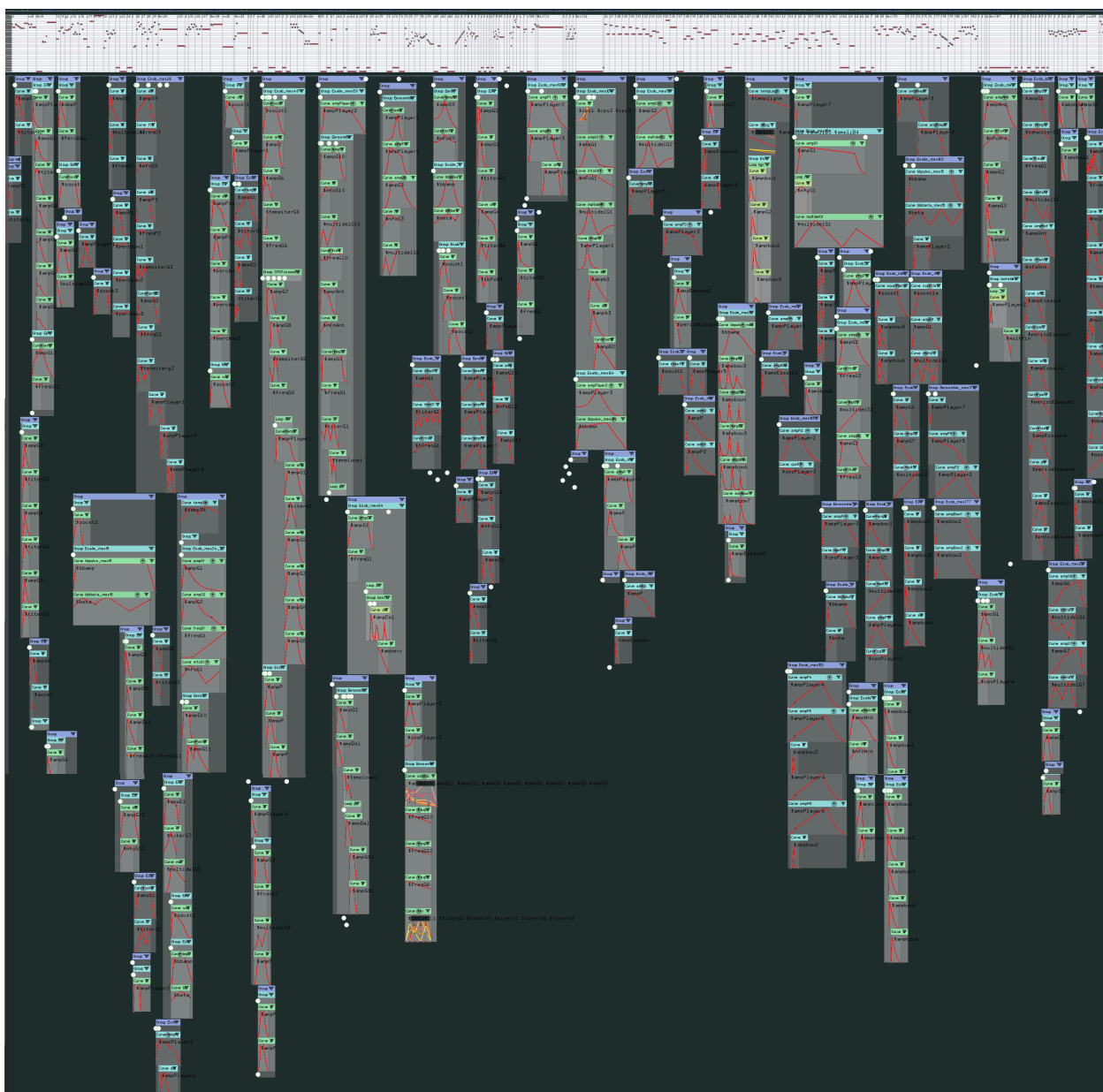
Dans une partition d'interprétation en revanche, une notation proche d'une certaine réalité acoustique « imagée » sera plus justifiée, de façon à ce qu'elle apporte les représentations les plus pertinentes pour l'instrumentiste qui devra jouer avec l'électronique. L'exemple historique de la pièce *Kontakte* de Karlheinz Stockhausen est évidemment un bon exemple. Par ailleurs, la diffusion de l'électronique dans les haut-parleurs pendant la performance, qu'elle soit faite par le compositeur, le RIM ou l'ingénieur du son est une partie extrêmement importante de l'interprétation d'une pièce mixte. Nous devons donc penser à une notation particulière pour cette utilisation précise, qui différera largement des deux autres types de notation. Sur ce sujet, nous renvoyons le lecteur à un article récent écrit par Marco Stroppa et Carlo Laurenzi sur le sujet de la notation des niveaux de dynamiques<sup>13</sup>.

Nous pouvons ici faire un parallèle avec les différences de styles de programmation en informatique. Un langage impératif ou procédural est focalisé sur l'état d'un programme ainsi que les fonctions et actions à réaliser pour obtenir un certain résultat. Un langage déclaratif se focalise davantage sur la spécification du résultat attendu et non sur la manière dont le calcul devra être réalisé. On a ici un lien direct avec nos concepts de partition de composition et de réalisation. La partition de composition s'exprimant généralement dans un domaine déclaratif quand la partition de réalisation s'établit elle dans un style de langage plus impératif. Si les deux styles de programmation (et également ici de notation) peuvent amener à un même résultat, le choix de l'utilisation de l'un ou de l'autre est lié à l'expressivité du langage, adapté au contexte dans lequel il s'inscrit.

## Représentations graphiques dans Antescofo

La musique électroacoustique et *a fortiori* la musique mixte impliquent des notations capables de représenter des éléments continus et discontinus. Lorsqu'il faut les intégrer dans un même espace, l'entreprise peut s'avérer périlleuse. Dans l'image présentée en page suivante, on voit la représentation graphique de la partition électronique (actions déclenchées en temps réel) d'une partie de *Namenlosen* dans le logiciel AscoGraph (interface graphique d'Antescofo). Dans cet exemple, seules les actions continues n'étant pas intégrées à des processus sont affichées (la partie claire en haut est un pianoRoll, représentant la partition du clavier suivi par Antescofo), ce qui signifie qu'un nombre extrêmement important d'informations et d'actions électroniques sont masquées. À l'inverse, l'image suivante n'affiche que les messages et envois de processus de façon à masquer les différentes courbes. Les deux images sont donc complémentaires. Mais même couplée, ces deux images masquent toutes les actions lancées par les processus qui ici ne sont représentés que par une ligne de message. D'une certaine manière, nous ne voyons que le premier niveau de la

<sup>13</sup> Carlo Laurenzi et Marco Stroppa, *The notation of dynamic levels in the performance of electronic music*. Proceedings of Tenor Conference, 2015.

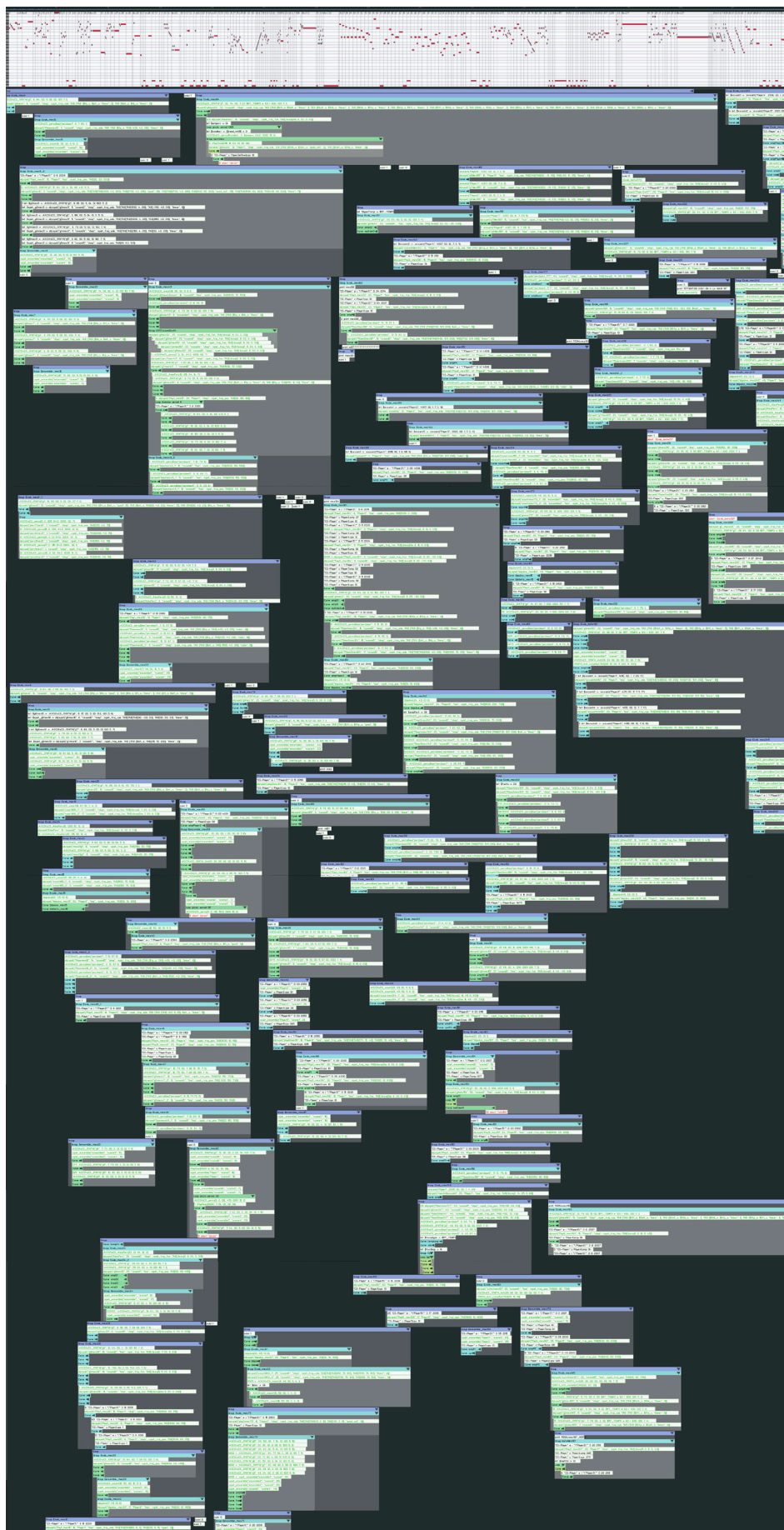


Représentation graphique d'un extrait de la partition Antescofo de *Namenlosen*. Seules les courbes sont affichées.

partition. La représentation dans cet exemple, semble pourtant très chargée alors même qu'elle ne montre qu'une vision très partielle des actions électroniques.

La question de la représentation des processus « hors-temps », déclenchés dynamiquement et évoluant selon des variables extérieures (notamment par l'utilisation d'un *whenever*, d'un *setvar* ou d'une valeur entrée par OSC) exige l'invention de nouveaux types de notation, pouvant éventuellement devenir dynamiques. Une simulation graphique de l'ensemble des actions déclenchées, qu'elles soient continues ou discontinues seraient une première étape.

La partition textuelle d'Antescofo est une partition de réalisation. L'ensemble des processus, des macros et des objets y sont définis. La partition électronique est greffée à la partition « réduite » — puisqu'elle n'affiche que des hauteurs et des durées — du musicien. Nous pouvons réaliser l'électronique de façon très fine, avec toutes les subtilités temporelles dont nous avons parlé précédemment. Cependant, les limites de la représentation graphique posent un problème compositionnel. Les représentations ne sont pas des simples retours visuels, elles nous permettent de réfléchir aux objets sur lesquels nous travaillons. Elles nous donnent souvent un nouveau *point de vue*, capable de remettre en cause une idée mal pensée ou de



Représentation graphique d'un extrait de la partition Antecofo de *Namenlosen*. Seules les actions atomiques et les groupes sont affichées.

nous montrer la non pertinence d'un processus. Les fonctions d'Antescofo telles que le *whenever* peuvent nous permettre de composer des environnements réactifs non dépendant du suivi d'une partition. En observant une ou plusieurs variables extérieures, on peut créer des processus qui réagiront à ces variables et à leur « temps ». Dans ces cas là la partition ne peut se lire linéairement, elle est un programme dans lequel le temps est, d'une certaine manière, implicite. Pour ce type de procédés, il n'existe pas de représentation — par extension de notations — autres que textuelles. Le manque de représentations visuelles limite donc, en partie, la notation de composition.

Malgré ce constat, les logiciels offrant des représentations visuelles (outils d'aide à la composition, interfaces graphiques telles que Max, Bach, Live...) restent beaucoup moins puissants en terme d'expressivité — de possibilités de composition de processus complexes et permettant une écriture fine *des temps* — que le langage textuel tel que celui d'Antescofo. Car si des milliers de lignes de codes peuvent paraître beaucoup moins attrayants qu'un ensemble de représentations graphiques, de courbes et d'illustrations, elles offrent paradoxalement un degré de liberté bien plus grand. Voilà pourquoi si nous utilisons des représentations graphiques à l'avenir, elles devront impérativement être ouvertes et programmables comme l'est un langage informatique, de façon à ce que l'imaginaire des compositeurs qui les *pratiquent* ne soient pas limité à celui des personnes qui l'auront programmée.

*« Voilà vingt ans que je répète que la maîtrise d'un ordinateur ne s'acquiert pas en quelques jours. C'est comme un instrument de musique : ça ne s'utilise pas, ça se pratique. Et dans la singularité de l'instrument, on va découvrir la singularité de sa propre pratique. La pratique ouvre à l'expérience, alors que la consommation ne produit que le conditionnement. »<sup>14</sup>*

En terme de notation, la musique mixte n'a pas encore vraiment dépassé le stade de la pierre de Rosette. La diversité des approches est telle, que l'on est en droit de se demander si une notation commune à tous serait non seulement possible, mais souhaitable. Quoiqu'il en soit, la question de la notation est sans nul doute l'une des plus importantes recherches musicales à mener dans le futur. Sa résolution, si elle est aujourd'hui personnelle à chaque compositeur, reste certainement la seule barrière qui puisse encore limiter leur pleine créativité.

La question qu'on peut poser pour clore ce chapitre est celle de savoir si les trois types de notation dont nous avons parlé — de composition, de réalisation et d'interprétation — pourraient se concilier au sein d'un même langage. Ces trois notations représentent en quelque sorte trois vues d'un même « objet ». On pourrait donc penser qu'un seul et même langage puisse les intégrer toutes. Or, le problème est qu'ici, nous avons au moins une notation qui ne peut s'incarner d'aucune manière dans un langage textuel et qui nécessite fondamentalement une notation graphique qui puisse se juxtaposer à une partition instrumentale : la notation d'interprétation.

Les deux autres notations (de composition et d'interprétation) peuvent s'accommoder l'une de l'autre et l'état actuel du langage Antescofo en est une version précaire, notamment par l'utilisation de fonctions qui permettent de rendre la partition plus lisible (macros, processus, objets, nims...) et par le filtrage possible d'informations. Mais nous sommes encore loin d'avoir une notation de composition qui nous permette d'appréhender aisément un processus complexe ou des superpositions de tempi dépendant de variables ou de calculs de façon convaincante. Peut-être faut-il s'intéresser aux développements qui ont été fait autour d'environnements de programmation comme Xcode<sup>15</sup>, notamment concernant les différentes fonctionnalités graphiques et de débogage.

<sup>14</sup> Bernard Stiegler, *Sauvons le capitalisme*. Technikart, novembre 2005.

<sup>15</sup> Xcode est un IDE (Integrated Development Environment), c'est un paquet d'outils qui facilitent la programmation. On y retrouve systématiquement un éditeur de texte, un compilateur et un débogueur.

# AU-DELÀ DE LA RÉTROACTION

*« Une atmosphère est quelque chose qui découle de l'idée centrale de l'auteur. Plus la formulation de cette idée centrale est fidèle, plus le sens de l'action est clair, et plus l'atmosphère tout autour sera signifiante. Les objets, les paysages, les intonations des acteurs, commenceront tous alors à résonner de cette note centrale. Tout deviendra interdépendant et indispensable. Chaque chose fera écho à une autre, tout s'interpellera, et il en résultera une atmosphère, comme conséquence de cette capacité à se concentrer sur le principal. Alors que vouloir créer une atmosphère en tant que telle, me paraît une chose étrange. »<sup>16</sup>*

L'un des reproches que l'on peut faire à l'électronique dans le contexte d'une musique mixte utilisant le temps-réel est qu'elle peut rapidement devenir une figuration triviale d'un rapport unilatéral et exclusivement causal. Autrement dit, l'électronique, liée en permanence au jeu du musicien, ne deviendrait qu'un accompagnement souple, intelligent dans son suivi mais peu dans son discours et, finalement, dans son rapport à l'instrument. Le problème est ici exclusivement compositionnel. Une plus ou moins grande complexité technologique peut accoucher d'effets dont la fonction ressemble davantage à l'utilisation de la pédale chez le pianiste débutant. L'écriture électronique est dans ces cas là évidemment assez grossière. Elle l'est dans son rapport unilatéral vis-à-vis du ou des interprètes mais aussi dans sa conception de l'électronique en elle-même, devenue simple productrice de son venant couvrir les éventuels vides de l'écriture instrumentale.

Nous ne polémiquerons pas davantage sur cette question. Nous souhaitons ici mettre en avant la conviction qu'il est désormais possible d'atteindre la même richesse d'écriture dans la partie électronique que dans la partie instrumentale. Les différentes possibilités d'écritures temporelles et de synchronisations peuvent nous permettre d'écrire des « voix » électroniques pouvant jouer avec un interprète humain mais n'en étant pas pour autant la simple conséquence ou le faire-valoir. Cela implique pour les compositeurs une connaissance relativement profonde des langages et des outils qu'ils voudront utiliser. La liberté gagnée par l'utilisation d'un langage puissant, qui permet d'amener l'écriture électronique à un niveau élevé, se paie en temps d'apprentissage et de conception.

La question de l'écriture de l'électronique n'a pas attendu le suivi de partition et le temps-réel pour être posée. Elle était déjà présente dans les premières pièces mixtes, on pense évidemment aux premières pièces mixtes de Stockhausen. Mais cette question doit être remise sur l'établi encore davantage aujourd'hui, dans un contexte où les outils de type « prêt-à-composer » font florès auprès des compositeurs (Live, Max for Live, Reason...). Il n'est plus très difficile de coller des sons ou des effets les uns après les autres, d'y superposer une voix instrumentale et de « créer des atmosphères ». La richesse d'écriture, de l'électronique en elle-même et des rapports qu'elle entretient avec un ou plusieurs interprètes, ne viendra qu'avec la volonté d'inscrire sa conception et sa réalisation à l'intérieur même de l'espace compositionnel au sens large. Peut-on seulement imaginer une œuvre faite de matière sans discours ? Le projet compositionnel s'élabore autour de la co-construction d'un outil devenu instrument et d'un langage capable de s'y adresser de façon convaincante. La partie réactive d'Antescofo, la richesse du tissu temporel qu'il permet de mettre en œuvre est selon nous capable d'assurer ce rôle de langage mais également de participer à la constitution de rapports véritablement dialectiques entre un instrument acoustique, joué par un musicien, et une partie électronique.

<sup>16</sup> Andreï Tarkovski, *Le temps scellé*. § *La responsabilité de l'artiste*. Éditions de l'étoile, Cahiers du cinéma, 1989.

## Organiser le multiple pour organiser le temps ?

Revenons ici à notre sujet liminaire — celui de la multiplicité — appréhendé cette fois dans le contexte de la musique mixte et de la notation de l'électronique. Nous avons vu qu'Antescofo mettait à disposition bon nombre de moyens « d'organisation de paramètres », que ce soit par des structures de données (tableaux, dictionnaires, nims) ou par l'utilisation de variables ajoutant un niveau symbolique supplémentaire et permettant dans certains cas l'explicitation des liens existants entre plusieurs couches, temporelles ou non. C'est à nous, compositeurs, qu'il appartient d'en faire bon usage, à savoir de donner un *sens* musical à chaque ensemble de paramètres. Pour un musicien, un symbole est bien plus porteur de sens qu'une suite de valeurs sans connexions évidentes. Décider qu'une variable \$brillant ou \$saturé ou \$profond ou \$f2m27 correspond à la suite de valeurs 3.24, 879, 47.2, 191 et 62 attribuée à tel module de synthèse ou de traitement ne relève que d'une seule chose : le choix, la prise de décision artistique, bref, l'émergence d'une singularité. Il existe aujourd'hui beaucoup de descripteurs nous donnant des indications sur les caractéristiques d'un son. Mais le geste du compositeur commence peut-être là où la description objective n'est pas congruente avec celle qu'il s'est construite, là où son imaginaire déborde du cadre et nécessite alors une refonte totale des catégories qu'il pensait jusque là immuables ou qu'un modèle d'analyse lui présentait comme indiscutable. Nous en revenons à notre idée centrale : la mise en relation. Ce sont les relations tissées entre divers éléments et leur association à une idée « d'échelle supérieure » qui donne un sens à l'ensemble.

La notion de qualité de temps dont nous avons parlé en première partie avait, entre autres choses, pour particularité de se définir à travers un réseau de paramètres en évolution sur plusieurs échelles de temps. Chaque échelle comprend une certaine mise en relation et l'ensemble des échelles lui-même est un réseau de dépendances. Nous avons exposé plus avant les différentes échelles de temps avec lesquelles nous pouvons composer grâce au langage Antescofo. Il s'agit d'amener l'écriture de l'électronique à un niveau qui permette la conjugaison de l'ensemble de ces échelles de temps. Si Antescofo n'est pas, en l'état actuel des choses, un environnement d'aide à la composition qui puisse être un lieu d'édification d'une pensée musicale (notamment en raison des faiblesses en terme de représentation), il est pour nous le moyen de la mettre en œuvre, de réaliser dans la partie électronique ce qui aura été conceptualisé dans l'espace compositionnel.

Temps des déclenchements, des modulations de signal, des enveloppes, du phrasé, des marqueurs formels et des processus globaux doivent dès lors s'entrelacer. Il ne s'agit pas seulement de transformer des pulsations en secondes, d'assurer un suivi de l'instrumentiste et d'y accoler un « accompagnement » avec une plus ou moins grande virtuosité. Il y a un enjeu *esthétique* majeur à ne plus considérer un langage comme celui d'Antescofo comme un simple moyen de superposer des sons — qu'ils soient électroniques ou concrets — au jeu d'un instrumentiste. Si une vraie notation musicale est aujourd'hui manquante, nous pouvons en revanche accéder à un niveau de finesse en terme d'*écriture* et notamment d'écriture des différents temps qui n'a jamais été aussi poussé. Les technologies, au sens large, ne sont plus « nouvelles », elles font éminemment partie de nous. Voilà pourquoi il faut sortir de cette image d'Épinal d'une musique mixte qui, mettant en avant la technologie en elle-même, se prosternant parfois devant ses prouesses, en oublie qu'elle n'est qu'un moyen et non une fin en soi. La technologie n'a pour nous d'intérêt que si elle est porteuse d'une « connaissance », qu'elle peut être un « lieu de pensée » dans lequel cette connaissance s'altère à notre contact, dans lequel puisse s'inscrire une vraie démarche compositionnelle et que de ce fait elle puisse être en mesure d'étendre le domaine de l'*écriture* musicale.

Le langage réactif d'Antescofo doit peut-être « s'émanciper » de la machine d'écoute. Si nous émettons cette hypothèse, c'est que nous pensons que les possibilités d'écriture qu'offre ce langage vont bien au-delà de la question du suivi de partition. Nous l'avons vu tout au long de cette partie, le langage en

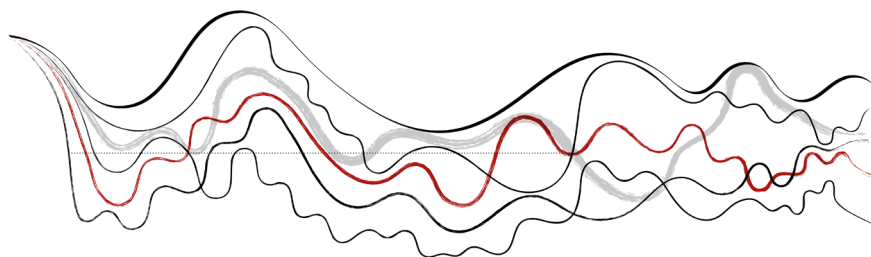


lui-même est un fabuleux outils d'écriture, et donc, peut tout à fait servir à une musique purement électronique qui ne mettrait pas en jeu le suivi de partition. Ce type d'utilisation est déjà présent aujourd'hui. Le langage en lui-même — appelons-le temporairement *Chronos* — pourrait constituer un noyau indépendant sur lequel pourrait se greffer Antescofo devenu alors pure machine d'écoute, mais aussi un futur module de notation — celui dont nous rêvons tous ! — un outils de suivi de geste, un ensemble de descripteurs, un module de synthèse ou tout autre outils pouvant être piloté depuis l'extérieur.

*Au commencement était le temps.*

Si il existe un continuum entre forme et matériau, alors nous devons nous emparer de tous les moyens en notre possession qui nous permettront d'imaginer, de réfléchir et de composer les entrelacements temporels qui le constitue. Entre un quatuor moyen et un très bon quatuor à cordes, il y a une différence fondamentale. Les interprètes du premier jouent dans un temps commun, conviennent de se retrouver sur tel ou tel point pivot et tentent d'assurer une synchronisation temporelle satisfaisante. Si nous voulions donner une image, nous aurions alors quatre lignes droites et parallèles traversées par quelques perpendiculaires de synchronisation subsumées par un temps-référence. Les interprètes du second forment quant à eux quatre lignes de temps singulières qui s'enchevêtrent, s'éloignent pour se rapprocher par la suite. Ils ne forment pas quatre individualités contrôlées par un temps tout-puissant d'échelle supérieure. Ils constituent cette intrication paradoxale qui consiste à ce que quatre individus ayant chacun leur propre temps fassent émerger de cet enchevêtrement de lignes, par les relations immanentes au discours musical qu'ils font vivre, un temps commun à tous, résultante de cette multiplicité.

*« Au soir du premier jour de combat,  
on vit en plusieurs endroits de Paris,  
au même moment et sans concertation,  
des gens tirer sur les horloges. »<sup>17</sup>*



<sup>17</sup> Walter Benjamin, *Thèses sur le concept d'histoire*.

# CONCLUSION

*« Et chacun a ses raisons, tout en se demandant ce qu'elles valent,  
et si ce sont les bonnes, pour aller là où il va plutôt qu'ailleurs, plutôt que nulle part. »<sup>1</sup>*

---

1 Samuel Beckett. *Malone Meurt*. Les Éditions de Minuit.

Si l'espace compositionnel en lui-même n'est pas neutre — il permet un repérage, il rend compte d'une géographie qui explicite des directions et donne un « sens » — sa constitution en revanche, ne l'est pas : elle s'accommode de diverses approches et méthodes, accueille toutes les formes de travail que peut choisir le compositeur : en spirale ou en expansion, du détail local à la forme globale, ou, tout au contraire de façon inductive, par raffinement progressif. La représentation spatiale est l'image de la singularisation, de l'organisation, de la structuration du matériau. Elle figure les nombreux liens existants en son sein. De ce fait l'espace compositionnel, par sa réification des rapports de proximité et de voisinage entre cellules musicales, expose au compositeur les rapports qui existent *de facto* à l'intérieur de son propre matériau. À partir de cela, il nous faut pouvoir passer d'une échelle à l'autre, organiser leurs plus ou moins grandes dépendances et voir comment l'ensemble des agencements de qualités temporelles sera en mesure de faire émerger une forme. Voilà pourquoi la notion de *pensée relationnelle* s'est avérée primordiale dans notre recherche, que ce soit dans nos représentations topologiques ou dans la pratique plus concrète d'un langage synchrone impliquant des processus parallèles.

L'environnement dans lequel nous avons implémenté nos espaces topologiques est loin d'être suffisant en terme de manipulation, d'expressivité et d'utilisation. Il nous faudra poursuivre le travail sur l'implémentation du modèle afin d'en faciliter la manipulation si nous voulons rendre plus souple la constitution d'un espace compositionnel. Les paramètres qui composent les cellules dont nous avons parlé ont été réévalués à chaque nouvelle création d'œuvre et le seront très certainement encore dans nos futures recherches. Le modèle, en revanche, s'est toujours avéré selon nous pertinent dans la mesure où il permettait une confrontation directe aux multiples hypothèses esthétiques qu'une pièce en cours d'élaboration pouvait engager.

Pour ce qui est du langage Antescofo, notre recherche a été limitée dans la mesure où elle n'a pas abordé la question cruciale de la notation. Si nous avons parlé de la richesse du langage, en terme d'expressivité temporelle et de la multitude de possibilités d'agencements de processus et de spécifications de *fluence* de l'électronique par le biais des stratégies de synchronisation, il apparaît clairement qu'un système de notation permettant de les représenter permettrait de passer un nouveau cap concernant l'écriture de l'électronique. Cette notation amènerait notamment avec elle la possibilité de pouvoir composer plus aisément des parties électroniques qui ne seraient pas directement liées au suivi d'une partition. Si il est déjà possible de composer des processus allant dans ce sens aujourd'hui, la possibilité d'en avoir un retour visuel permettrait selon nous une avancée majeure en terme de finesse d'écriture et de pensée de ces processus.

Les fenêtres temporelles dont nous avons brièvement parlé n'ont pu être testées en situation de concert et méritent un vrai approfondissement — autant d'un point de vue technique que compositionnel — pour en percevoir toute la fécondité. Nous pensons n'avoir qu'effleuré les potentialités que pourraient engendrer cette idée de fenêtre temporelle, notamment dans le domaine qui nous concerne, à savoir les multiplicités temporelles et leurs interrelations en musique mixte. Les prochaines créations seront, entre autres choses, l'occasion de focaliser notre recherche sur cet aspect particulier du langage.

## CONCLUSION

Si les technologies audionumériques sont aujourd'hui accessibles au plus grand nombre, si elles font résolument partie de notre environnement quotidien, il y a un enjeu majeur à ce que la connaissance de ceux qui les utilisent soit suffisamment grande à leurs sujets pour qu'ils n'en soient pas les simples consommateurs, mais qu'au contraire — et encore davantage s'il s'agit de compositeurs — ils en soient des pratiquants *critiques*. La musique mixte implique une *pratique* pour que l'utilisation de la technologie qu'elle implique puisse dépasser les simples enjeux techniques et s'inscrive indubitablement dans le discours et plus généralement la pensée musicale du compositeur. Si les logiciels et langages sont faits par des informaticiens, les représentations et modes de pensée qui en découlent ne coïncident pas toujours avec ceux des compositeurs. Là est certainement l'une des grandes forces d'un institut comme l'Ircam, faisant cohabiter scientifiques et compositeurs dans un même espace.

« *Énormément de travail, beaucoup de méthode et de soins méticuleux sont nécessaires pour parvenir au laisser-faire : c'est le moment où la matière se délivre d'elle-même et où les choses se donnent dans leur fugue.* »<sup>2</sup>

\* \* \*

L'ensemble des outils développés durant cette thèse, qu'il s'agisse des espaces compositionnels ou du langage Antescofo font aujourd'hui partie intégrante de mon travail de création. Il n'a pas toujours été évident de conjuguer la recherche et la composition. Cependant, la pratique a eu l'immense avantage de mettre à l'épreuve les hypothèses de recherche et d'ébranler la moindre de mes certitudes.

Mon travail sur l'écriture électronique et sur la multiplicité des temps me semble, après ces trois années de recherche, n'en être paradoxalement qu'à son commencement. Les partitions électroniques que j'ai réalisé pour mes différentes pièces ont beaucoup évolué si l'on regarde la première, *Tesla*, et la dernière en date, *Namenlosen*. La notation est devenue bien plus expressive et concise, laissant donc plus de place au « sens » musical. La route est malgré tout encore bien longue, avant d'arriver à une notation qui puisse s'intégrer à celle des instrumentistes ou qui simule des processus complexes mettant en jeu une multitude de temps.

On pourrait se demander si l'ensemble des « architectures », des structures imbriquées ou des finesses d'écritures mises en place peut vraiment se révéler à l'auditeur écoutant l'œuvre qui en résulte. C'est d'ailleurs une question qui devient récurrente aujourd'hui. Je reste persuadée que même si la réalité vient souvent frapper à la porte de façon plus ou moins contrariante, il reste toujours *quelque chose qui suinte à la surface*.

Les espaces compositionnels que j'ai créé et que je continuerai à développer à l'avenir, ont été justement pensés à travers ce postulat esthétique. La pensée relationnelle à laquelle j'ai souvent fait référence me semble importante pour l'avenir, dans la mesure où elle rend *lisible* les processus dialectiques inhérents à la création musicale. Il me semble que cette notion de dialectique a souvent été envisagée comme une résultante, que ce soit d'un certain degré de complexité ou des rapports immanents à un matériau sonore. Je crois que cette manière de voir la dialectique est quelque peu « passive ». Il serait temps, en tout cas dans la démarche qui est la mienne, de *pratiquer* une dialectique non par hasard, mais par *volonté*.

---

<sup>2</sup> Valère Novarina, *Devant la parole*. Éditions P.O.L., 2010.

## CONCLUSION

*« Sans doute le feu s'est-il retiré,  
l'incendie maîtrisé,  
mais s'il y a là cendre,  
c'est que du feu reste en retrait. »<sup>3</sup>*

---

<sup>3</sup> Jacques Derrida, *Feu la cendre*. Éditions des femmes, Antoinette Fouque, 1987.



# ANNEXE I



IMPLÉMENTATION MATHEMATICA  
DES ESPACES DE *NAMENLOSEN*

# Complexes simpliciaux — construction

Implémentation de la fonction de construction du complexe simplicial :

```
ClearAll[Complex2Graph, Plonge, Plonge2, PlongeMethod, g, aux1, Export2Autocad]

Complex2Graph[l_List] := Graph[Cases[l, Simplex[x_, y_] /. Simplex -> DirectedEdge]

PlongeMethod[n_] :=
Take[{"BipartiteEmbedding", "CircularEmbedding", "CircularMultipartiteEmbedding", "DiscreteSpiralEmbedding",
  "GridEmbedding", "LinearEmbedding", "MultipartiteEmbedding", "SpiralEmbedding", "StarEmbedding",
  "BalloonEmbedding", "RadialEmbedding", "LayeredDigraphEmbedding", "LayeredEmbedding", "HighDimensionalEmbedding",
  "PlanarEmbedding", "SpectralEmbedding", "SpringElectricalEmbedding", "SpringEmbedding"}, {n}]

aux1[x_, {n_}] := (x -> n)

(* sélecteur de voisins d'un simplexe (SOMMETS) *)
selVoisins[l1_List, p_List, filt_, vertexl_List] :=
Extract[p, #] & /@ Flatten[Position[vertexl, #, 1] & /@ Union[Flatten[Cases[l1, #] & /@ Permutations[filt], 2]]]

(*-----*)

Plonge2[l2_List, lstr_List, x_Integer:1, filt_List, filts_List] :=
With[{
  g = Graph[(* Pour transformer n'importe quelle liste en paire de 2... *)
    (DeleteDuplicates[Flatten[Map[Subsets[#, {2, 2}] &, l2], 1]]
    /. {a_, b_} -> UndirectedEdge[a, b]]],
  With[{vl = VertexList[g], e = EdgeList[g], vstr = DeleteDuplicates[Flatten[Map[Subsets[#, {2, 2}] &, lstr], 2]],
    p = GraphEmbedding[g, PlongeMethod[x], 3]],
  With[{r = MapIndexed[aux1, vl], r2 = MapIndexed[aux1, Union[Flatten[Cases[l2, #] & /@ Permutations[filt]]]],
  With[{data2 = l2 /. r, data3 = (Flatten[Cases[l2, #] & /@ Permutations[filt], 1]) /. r2},

  Manipulate[
  Graphics3D[
  {GraphicsComplex[p, {
    {Black, PointSize[Large], Point[p]},
    Map[Thread[Text[#, p]] &,
      {Delete[DeleteDuplicates[Flatten[Map[{Text[If[StringMatchQ[#, RegularExpression["a\\w*"]],
        Style[Symbol[#, 12, Bold, Orange, Opacity[1], Background -> Transparent]]],
        Text[If[StringMatchQ[#, RegularExpression["t\\w*"]], Style[Symbol[#, 12, Bold, Green,
          Opacity[1], Background -> Transparent]]],
        Text[If[StringMatchQ[#, RegularExpression["f\\w*"]], Style[Symbol[#, 12, Bold, Cyan,
          Opacity[1], Background -> Transparent]]]} &, vstr]], 2]]},
    Opacity[opacité], Lighting -> Automatic, Polygon[data2]]},

  (* Partie tradatée de la structure sélectionnée par le voisinage du simplexe*)
  Translate[GraphicsComplex[selVoisins[l2, p, filt, vl],
    {{Black, PointSize[Large], Point[selVoisins[l2, p, filt, vl]]},
    Map[Thread[Text[#, selVoisins[l2, p, filt, vl]]] &,
      {Delete[DeleteDuplicates[Flatten[Map[{Text[If[StringMatchQ[#, RegularExpression["a\\w*"]],
        Style[Symbol[#, 12, Bold, Orange, Opacity[1], Background -> Transparent]]],
        Text[If[StringMatchQ[#, RegularExpression["t\\w*"]], Style[Symbol[#, 12, Bold, Green,
          Opacity[1], Background -> Transparent]]],
        Text[If[StringMatchQ[#, RegularExpression["f\\w*"]], Style[Symbol[#, 12, Bold, Cyan,
          Opacity[1], Background -> Transparent]]]} &,
        Union[Flatten[Cases[lstr, #] & /@ Permutations[filts], 2]]]], 2]]},
    Opacity[opacité], Lighting -> Automatic, Polygon[data3]]], {Z, Z, Z}}
  , ImageSize -> {1250, 1050}, Boxed -> False, Background -> Black], {Z, 0, 3},
  {opacité, 0., 1.},
  Initialization -> (opacité = 0.5)
  ]]]
]
]
```

# Namenlosen

```
AppendTo[$Path,
  ToFileName[{$HomeDirectory,
    «/Users/juliablondeau/Google\\ \
Drive/JBLO_PERSO/JBLO_DOCTORAT/COMPOSITION/2016_NAMENLOSEN/ESPACES/\
MUND1\\ -\\ Signaux»}]];

ClearAll[Env1, Env2, Env3, Env4, Env5, Env6, Env7, Env8, Env9, Env10, \
Env11, Fenv1, Fenv2, Fenv3, Fenv4, Fenv5, Fenv6, Fenv7, Fenv8]
ClearAll[M1, M2, M3, M4, M5, M6];
viewCompA = 2;
viewCompT = 2;
viewCompF = 2;
viewCompG = 2;
viewEnv = 2;
Mv1 = 1;
Mv2 = 1;
Mv3 = 1;
Mv4 = 1;
Mv5 = 1;
Mv6 = 1;
```

## Bibliothèque d'enveloppes

```
NewEnvA[n_, ll_List] :=
  n = FlipView[{ToString[n],
    ListLinePlot[ll, Filling -> Bottom, FillingStyle -> Orange,
      Axes -> False, ImageSize -> 30]}, Dynamic[viewEnv]]
NewEnvF[n_, ll_List] :=
  n = FlipView[{ToString[n],
    ListLinePlot[ll, Filling -> Bottom, FillingStyle -> Cyan,
      Axes -> False, ImageSize -> 30]}, Dynamic[viewEnv]]

NewEnvA[Env1, {{0., 0.}, {1, 0.8}, {2, 0.6}, {20, 0.}}];
NewEnvA[Env2, {{0., 0.}, {6, 0.2}, {10, 0.5}, {11, 0.8}, {12,
  0.5}, {16, 0.2}, {22, 0}}];
NewEnvA[Env3, {{0., 0.2}, {1, 0.2}}];
NewEnvA[Env4, {{0., 0.}, {8, 0.8}, {8.1, 0.}}];
NewEnvA[Env5, {{0., 0.}, {1, 0.8}, {3, 0.2}, {18, 0.2}, {20,
  0.8}, {21, 0.}}];
NewEnvA[Env6, {{0., 0.}, {2, 0.1}, {4, 0.3}, {6, 0.5}, {8, 0.3}, {10,
  0.1}, {12, 0.}}];
NewEnvA[Env7, (Accumulate[RandomReal[{-1, 1}, 200]])];
NewEnvA[Env8, {{0., 0.}, {2, 0.05}, {4, 0.15}, {6, 0.4}, {8, 0.95}}];
NewEnvA[Env9, {{0., 1.}, {7, 0.01}, {9, 0.01}, {16, 1.}}];
NewEnvA[Env10, {{0., 0.}, {1, 1.}, {2, 0.2}, {30, 0.2}}];
NewEnvA[Env11, {{0., 0.}, {1, 1.}, {8, 1.}, {9, 0.}}];

NewEnvF[Fenv1, {{0., 0.8}, {20, 0.}}];
NewEnvF[Fenv2, {{0., 0.}, {6, 0.2}, {10, 0.5}, {11, 0.8}, {12,
  0.5}, {16, 0.2}, {22, 0}}];
NewEnvF[Fenv3, {{0., 0.2}, {1, 0.2}}];
NewEnvF[Fenv4, {{0., 0.}, {8, 0.8}}];
NewEnvF[Fenv5, (Accumulate[RandomReal[{-1, 1}, 200]])];
NewEnvF[Fenv6, {{0., 0.8}, {6, 0.5}, {10, 0.2}, {11, 0.}, {12,
```

```

0.2}, {16, 0.5}, {22, 0.8}}];
NewEnvF[Fenv7, {{0., 0.}, {2, 0.05}, {4, 0.15}, {6, 0.4}, {8,
0.95}}];
NewEnvF[Fenv8, {{0., 0.95}, {2, 0.4}, {4, 0.15}, {6, 0.05}, {8, 0.}}];

```

## Dictionnaire timbre

```
ViewA[n_] :=
```

```

n = FlipView[{ToString[n],
Labeled[ToExpression["t" <> ToString[n]], ToString[n]]},
Dynamic[viewCompA]];
ClearAll[t101, t102, t103, t104, t105, t106, t107, t108, t109, t110,
t111];
(ViewA[#]) & /@ {t101, t102, t103, t104, t105, t106, t107, t108, t109,
t110, t111};
ClearAll[t201, t202, t203, t204, t205, t206, t207, t208, t209, t210,
t211, t212];
(ViewA[#]) & /@ {t201, t202, t203, t204, t205, t206, t207, t208, t209,
t210, t211, t212};
ClearAll[t301, t302, t303, t304, t305, t306, t307, t308, t309, t310,
t311, t312];
(ViewA[#]) & /@ {t301, t302, t303, t304, t305, t306, t307, t308, t309,
t310, t311, t312};
ClearAll[t401, t402, t403, t404, t405, t406, t407, t408, t409, t410,
t411, t412, t413];
(ViewA[#]) & /@ {t401, t402, t403, t404, t405, t406, t407, t408, t409,
t410, t411, t412, t413};

```

```
tt0 = {"enveloppe", "dynamique", "timbre 1", "timbre 2 / d. hétéro"};
```

```

tt101 = {Env3, «mp», «r», «e/h-nul»};
tt102 = {Env4, «mf», «n», «r/h-faib»};
tt103 = {Env11, «f», «n», «s/h-moy»};
tt104 = {Env3, «p», «r-e», «b/h-moy»};
tt105 = {Env2, «mf», «e», «s/h-faib»};
tt106 = {Env6, «f», «n», «r/h-nul»};
tt107 = {Env10, «mp», «e», «r/h-faib»};
tt108 = {Env3, «p», «r», «e/h-moy»};
tt109 = {Env7, «mp», «r-m», «n/h-fort»};
tt110 = {Env2, «mf», «e», «s/h-faib»};
tt111 = {Env1, «f», «r», «m/h-faib»};

```

```

tt201 = {Env3, «pp», «r», «r/h-moy»};
tt202 = {Env3, «p», «e», «r/h-faib»};
tt203 = {Env3, «p», «r», «n/h-moy»};
tt204 = {Env10, «mp», «e», «r/h-moy»};
tt205 = {Env10, «pp», «n», «r/h-faib»};
tt206 = {Env7, «p», «m», «b/h-fort»};
tt207 = {Env7, «p», «r», «n/h-faib»};
tt208 = {Env2, «mp», «e», «r/h-moy»};
tt209 = {Env5, «pp», «r», «n/h-faib»};
tt210 = {Env3, «pp», «b», «r/h-moy»};
tt211 = {Env6, «p», «b», «r/h-moy»};
tt212 = {Env10, «p», «b», «e/h-moy»};

```

```

tt301 = {Env11, «f», «n-e», «s/h-moy»};
tt302 = {Env11, «ff», «s», «e/h-faib»};
tt303 = {Env4, «mf», «e», «r/h-faib»};

```

## ANNEXE 1

```

tt304 = {Env8, «mf», «e», «s/h-faib»};
tt305 = {Env7, «mf», «e», «n/h-moy»};
tt306 = {Env1, «mp», «n», «e/h-moy»};
tt307 = {Env6, «ff», «s», «e/h-fort»};
tt308 = {Env11, «f», «e», «r/h-faib»};
tt309 = {Env8, «f», «n», «e/h-moy»};
tt310 = {Env6, «f», «n», «s/h-faib»};
tt311 = {Env11, «ff», «s», «e/h-fort»};(*pour les blocs...*)

tt401 = {Env6, «mp», «e», «s/h-faib»};
tt402 = {Env6, «mf», «r», «n/h-faib»};
tt403 = {Env2, «f», «r», «s/h-moy»};
tt404 = {Env2, «mp», «r», «e/h-moy»};
tt405 = {Env3, «p», «r», «e/h-faib»};
tt406 = {Env3, «p», «m», «r/h-moy»};
tt407 = {Env9, «mf», «n», «r/h-faib»};
tt408 = {Env7, «mp», «r», «m/h-fort»};
tt409 = {Env7, «mf», «e», «r/h-moy»};
tt410 = {Env1, «f», «r», «n/h-faib»};
tt411 = {Env5, «p», «r», «e/h-moy»};
tt412 = {Env6, «p», «b», «e/h-moy»};
tt413 = {Env10, «mp», «b», «r/h-faib»};

```

## Dictionnaire timbre

```

ViewT[n_] :=
  n = FlipView[{ToString[n],
    Labeled[ToExpression[«r» <> ToString[n]], ToString[n]],
    Dynamic[viewCompT]}];
ClearAll[r101, r102, r103, r104, r105, r106, r107, r108, r109];
(ViewT[#]) & /@ {r101, r102, r103, r104, r105, r106, r107, r108, r109};
ClearAll[r201, r202, r203, r204, r205, r206, r207, r208, r209];
(ViewT[#]) & /@ {r201, r202, r203, r204, r205, r206, r207, r208, r209};
ClearAll[r301, r302, r303, r304, r305, r306, r307, r308, r309, r310];
(ViewT[#]) & /@ {r301, r302, r303, r304, r305, r306, r307, r308, r309,
  r310};
ClearAll[r401, r402, r403, r404, r405, r406, r407, r408, r409, r410,
  r411];
(ViewT[#]) & /@ {r401, r402, r403, r404, r405, r406, r407, r408, r409,
  r410, r411};

rr0 = {«métrique», «profil», «vitesse/striage», «régularité»};

rr101 = {«(4,•---)», «_», «v-lent», «r-fort»};
rr102 = {«(8,-•---•---)», «/\», «v-moy», «r-moy»};
rr103 = {«(8,-•---•---)», «/», «v-moy», «r-fort»};
rr104 = {«(6,•---•---)», «_», «v-lent», «r-moy»};
rr105 = {«(5,-••---)», «_», «v-lent», «r-faib»};
rr106 = {«(5,••••-)», «_», «v-rap»,
  «r-moy»};
rr107 = {«(6,-----)», «/\», «v-lent», «r-fort»};
rr108 = {«(9,•---•---•••)», «_»,
  «v-moy», «r-moy»};
rr109 = {«(7,••---••-)», «\»,
  «v-lent», «r-fort»};

rr201 = {«(6,•••••-)», «_»,
  «v-moy», «r-faib»};

```

## ANNEXE 1

```

rr202 = {«(5,•-0-•)», «_», «v-lent», «r-fort»};
rr203 = {«(6,--•-0-)», «\», «v-moy», «r-faib»};
rr204 = {«(9,-•••••--•-)»,
  «/\», «v-moy», «r-nul»};
rr205 = {«(4,-00-)», «_», «v-moy», «r-fort»};
rr206 = {«(6,-0•-0-)», «_», «v-trl», «r-moy»};
rr207 = {«(6,-0-0••)», «/», «v-rap», «r-faib»};
rr208 = {«(7,•--••0-)», «_», «v-moy»,
  «r-faib»};
rr209 = {«(9,•0••--0••)»,
  «\», «v-moy», «r-faib»};

rr301 = {«(8,--•••••-)», «_»,
  «v-moy», «r-fort»};
rr302 = {«(10,•--•-•••\
•-)», «_», «v-rap», «r-faib»};
rr303 = {«(7,•••-•--)», «\», «v-moy»,
  «r-faib»};
rr304 = {«(7,•-•••••)\
, «_», «v-rap», «r-moy»};
rr305 = {«(6,----•-)», «/\», «v-lent», «r-faib»};
rr306 = {«(6,•••••-)», «_»,
  «v-rap», «r-moy»};
rr307 = {«(8,-•--••-•)», «_», «v-moy»,
  «r-faib»};
rr308 = {«(11,-•-••••\
--•-)», «_», «v-moy», «r-faib»};
rr309 = {«(9,•--•--••-)», «/»,
  «v-rap», «r-faib»};
rr310 = {«(9,••-•-•••\
•)», «_», «v-trap», «r-moy»};

rr401 = {«(6,•----•)», «_», «v-moy», «r-faib»};
rr402 = {«(9,•--•--0--)», «/\», «v-lent», «r-faib»};
rr403 = {«(9,-•-•••••-\
)», «_», «v-rap», «r-moy»};
rr404 = {«(8,--•--•--)», «\», «v-moy», «r-fort»};
rr405 = {«(5,••••-)», «_», «v-rap»,
  «r-moy»};
rr406 = {«(9,•-•••••--)\
, «~», «v-moy», «r-faib»};
rr407 = {«(7,---•---)», «/\», «v-lent», «r-fort»};
rr408 = {«(7,•----•-)», «\», «v-rap», «r-faib»};
rr409 = {«(13,•••••-•-\
•••••)», «_», «v-trap»,
  «r-moy»};
rr410 = {«(10,•-•--•••\
••)», «/», «v-trap», «r-fort»};
rr411 = {«(11,-0•--•-••--)», «~»,
  «v-rap», «r-faib»};

```

Dictionnaire timbre

## Dictionnaire fréquentiel

ViewF[n\_] :=

```

n = FlipView[{ToString[n],
  Labeled[ToExpression[«f» <> ToString[n]], ToString[n]]},
  Dynamic[viewCompF]];

```

## ANNEXE 1

```

ClearAll[f101, f102, f103, f104, f105, f106, f107];
(ViewF[#]) & /@ {f101, f102, f103, f104, f105, f106, f107};
ClearAll[f201, f202, f203, f204, f205, f206, f207];
(ViewF[#]) & /@ {f201, f202, f203, f204, f205, f206, f207};
ClearAll[f301, f302, f303, f304, f305, f306, f307, f308, f309, f310];
(ViewF[#]) & /@ {f301, f302, f303, f304, f305, f306, f307, f308, f309,
  f310};
ClearAll[f401, f402, f403, f404, f405, f406, f407, f408];
(ViewF[#]) & /@ {f401, f402, f403, f404, f405, f406, f407, f408};

ff0 = {«matrice», «polarisation», «ambitus», «profil»};
ff101 = {M1, «p-moy», «a-gra», Fenv5};
ff102 = {M1, «p-nul», «a-gra», Fenv5};
ff103 = {M1, «p-moy», «a-moy», Fenv5};
ff104 = {M1, «p-gra», «a-gra», Fenv7};
ff105 = {M1, «p-nul», «a-gra», Fenv7};
ff106 = {M1, «p-fort», «a-red», Fenv3};
ff107 = {M1, «p-faib», «a-red», Fenv1};

ff201 = {M2, «p-nul», «a-moy», Fenv5};
ff202 = {M2, «p-fort», «a-red», Fenv3};
ff203 = {M2, «p-fort», «a-gra», Fenv3};
ff204 = {M2, «p-moy», «a-gra», Fenv5};
ff205 = {M2, «p-faib», «a-moy», Fenv3};
ff206 = {M2, «p-faib», «a-gra», Fenv2};
ff207 = {M2, «p-nul», «a-gra», Fenv5};

ff301 = {M3, «p-fort», «a-moy», Fenv3};
ff302 = {M3, «p-moy», «a-red», Fenv3};
ff303 = {M3, «p-faib», «a-gra», Fenv6};
ff304 = {M3, «p-moy», «a-moy», Fenv6};
ff305 = {M3, «p-nul», «a-gra», Fenv7};
ff306 = {M3, «p-faib», «a-moy», Fenv1};
ff307 = {M3, «p-faib», «a-gra», Fenv4};
ff308 = {M3, «p-moy», «a-gra», Fenv3};
ff309 = {M3, «p-nul», «a-moy», Fenv5};
ff310 = {M3, «p-nul», «a-red», Fenv5};

ff401 = {M4, «p-nul», «a-gra», Fenv2};
ff402 = {M4, «p-moy», «a-moy», Fenv2};
ff403 = {M4, «p-fort», «a-red», Fenv3};
ff404 = {M4, «p-moy», «a-moy», Fenv3};
ff405 = {M4, «p-nul», «a-gra», Fenv7};
ff406 = {M4, «p-fort», «a-gra», Fenv2};
ff407 = {M4, «p-nul», «a-gra», Fenv8};
ff408 = {M4, «p-faib», «a-moy», Fenv6};

```

## Dictionnaire des attractions

```

ViewG[n_] :=
  n = FlipView[{ToString[n],
    Labeled[ToExpression[«a» <> ToString[n]], ToString[n]],
    Dynamic[viewCompA]}];
ClearAll[a101, a102, a103, a104, a105, a106, a107, a108, a109, a110,
  a111, a112, a113, a114, a115, a116];
(ViewG[#]) & /@ {a101, a102, a103, a104, a105, a106, a107, a108, a109,
  a110, a111, a112, a113, a114, a115, a116};

```

## ANNEXE 1

```

ClearAll[a201, a202, a203, a204, a205, a206, a207, a208, a209, a210,
a211, a212, a213, a214, a215, a216];
(ViewG[#]) & /@ {a201, a202, a203, a204, a205, a206, a207, a208, a209,
a210, a211, a212, a213, a214, a215, a216};
ClearAll[a301, a302, a303, a304, a305, a306, a307, a308, a309, a310,
a311, a312, a313, a314, a315, a316];
(ViewG[#]) & /@ {a301, a302, a303, a304, a305, a306, a307, a308, a309,
a310, a311, a312, a313, a314, a315, a316};
ClearAll[a401, a402, a403, a404, a405, a406, a407, a408, a409, a410,
a411, a412, a413, a414, a415, a416];
(ViewG[#]) & /@ {a401, a402, a403, a404, a405, a406, a407, a408, a409,
a410, a411, a412, a413, a414, a415, a416};

```

```
aa0 = {«sym harmo», «sym rythm», «tension timbrique», «directivité»};
```

```

aa101 = {«sh-excl», «sr-excl», «rr-faib», «dir-nul»};
aa102 = {«sh-excl», «sr-moy», «rr-moy», «dir-moy»};
aa103 = {«sh-excl», «sr-moy», «rr-faib», «dir-faib»};
aa104 = {«sh-excl», «sr-faib», «rr-moyenne», «dir-moy»};
aa105 = {«sh-excl», «sr-excl», «rr-nul», «dir-nul»};
aa106 = {«sh-fort», «sr-moy», «rr-faib», «dir-faib»};
aa107 = {«sh-fort», «sr-moy», «rr-moy», «dir-moy»};
aa108 = {«sh-fort», «sr-faib», «rr-faib», «dir-fort»};
aa109 = {«sh-fort», «sr-faib», «rr-moy», «dir-fort»};
aa110 = {«sh-moy», «sr-fort», «rr-nul», «dir-fort»};
aa111 = {«sh-moy», «sr-fort», «rr-moy», «dir-faib»};
aa112 = {«sh-moy», «sr-faib», «rr-moy», «dir-moy»};
aa113 = {«sh-moy», «sr-moy», «rr-faib», «dir-fort»};
aa114 = {«sh-fort», «sr-fort», «rr-moy», «dir-moy»};
aa115 = {«sh-fort», «sr-moy», «rr-faib», «dir-faib»};
aa116 = {«sh-excl», «sr-excl», «rr-fort», «dir-nul»};

```

```

aa201 = {«sh-moy», «sr-faib», «rr-faib», «dir-fort»};
aa202 = {«sh-moy», «sr-moy», «rr-nul», «dir-faib»};
aa203 = {«sh-moy», «sr-fort», «rr-nul», «dir-fort»}; (* type signal *)
aa204 = {«sh-moy», «sr-faib», «rr-moy», «dir-faib»};
aa205 = {«sh-moy», «sr-faib», «rr-faib», «dir-moy»};
aa206 = {«sh-faib», «sr-moy», «rr-moy», «dir-fort»};
aa207 = {«sh-fort», «sr-nul», «rr-nul», «dir-faib»};
aa208 = {«sh-fort», «sr-nul», «rr-faib», «dir-faib»};
aa209 = {«sh-faib», «sr-fort», «rr-moy», «dir-fort»};
aa210 = {«sh-faib», «sr-faib», «rr-moy», «dir-fort»};
aa211 = {«sh-excl», «sr-nul», «rr-faib», «dir-faib»};

```

```

aa301 = {«sh-nul», «sr-faib», «rr-fort», «dir-fort»};
aa302 = {«sh-fort», «sr-moy», «rr-moy», «dir-fort»};
aa303 = {«sh-faib», «sr-faib», «rr-faib», «dir-fort»};
aa304 = {«sh-faib», «sr-moy», «rr-moy», «dir-faib»};
aa305 = {«sh-faib», «sr-nul», «rr-fort», «dir-moy»};
aa306 = {«sh-nul», «sr-fort», «rr-moy», «dir-moy»};
aa307 = {«sh-nul», «sr-moy», «rr-moy», «dir-moy»};
aa308 = {«sh-fort», «sr-faib», «rr-faib», «dir-moy»};
aa309 = {«sh-moy», «sr-faib», «rr-moy», «dir-fort»};
aa310 = {«sh-moy», «sr-nul», «rr-fort», «dir-fort»};
aa311 = {«sh-faib», «sr-moy», «rr-fort», «dir-fort»};
aa312 = {«sh-faib», «sr-nul», «rr-moy», «dir-faib»};

```

```

aa401 = {«sh-moy», «sr-nul», «rr-moy», «dir-fort»};
aa402 = {«sh-moy», «sr-moy», «rr-faib», «dir-fort»};

```



## ANNEXE 1

```
aa403 = {«sh-fort», «sr-fort», «rr-moy», «dir-moy»};
aa404 = {«sh-faib», «sr-faib», «rr-fort», «dir-fort»};
aa405 = {«sh-faib», «sr-fort», «rr-moy», «dir-fort»};
aa406 = {«sh-faib», «sr-moy», «rr-faib», «dir-moy»};
aa407 = {«sh-moy», «sr-faib», «rr-fort», «dir-moy»};
aa408 = {«sh-moy», «sr-moy», «rr-faib», «dir-faib»};
```

# Déclaration des espaces

(\* Déclaration de l'environnement \*)

```
Mund1 = {{t101, r101, f106, a108}, {t101, r101, f102, a113}, {t101, r104, f105, a110},
{t102, r106, f105, a109}, {t102, r108, f101, a103}, {t102, r104, f102, a111},
{t103, r107, f103, a104}, {t103, r105, f106, a105}, {t103, r101, f106, a114}, {t103, r103, f104, a102},
{t104, r109, f106, a109}, {t104, r107, f105, a101}, {t104, r103, f102, a105},
{t105, r102, f102, a113}, {t105, r102, f105, a109}, {t105, r108, f103, a111},
{t106, r107, f101, a106}, {t106, r108, f101, a115}, {t106, r101, f107, a114},
{t107, r101, f106, a106}, {t107, r104, f102, a103}, {t107, r108, f107, a107}, {t107, r101, f102, a101},
{t108, r107, f106, a114}, {t108, r109, f101, a102}, {t108, r109, f106, a111},
{t109, r102, f102, a111}, {t109, r105, f104, a103}, {t109, r103, f104, a101},
{t110, r105, f105, a114}, {t110, r103, f101, a116}, {t110, r107, f105, a116},
{t111, r109, f107, a108}, {t111, r102, f107, a109}, {t111, r103, f105, a113}};

Mund2 = {{t201, r201, f202, a201}, {t201, r206, f201, a202}, {t201, r203, f203, a208},
{t202, r202, f205, a209}, {t202, r209, f204, a211}, {t202, r201, f204, a201},
{t203, r208, f207, a205}, {t204, r208, f207, a211}, {t204, r209, f205, a209},
{t205, r202, f202, a203}, {t205, r204, f204, a207}, {t205, r204, f205, a202},
{t206, r209, f207, a211}, {t206, r206, f203, a204}, {t206, r203, f206, a204},
{t207, r204, f204, a202}, {t207, r208, f202, a201}, {t207, r203, f205, a208},
{t208, r204, f206, a209}, {t208, r206, f203, a205}, {t208, r209, f207, a211},
{t209, r209, f207, a208}, {t209, r208, f202, a207}, {t209, r204, f207, a205},
{t210, r205, f205, a205}, {t210, r201, f205, a210}, {t210, r207, f202, a209},
{t211, r206, f206, a210}, {t211, r205, f201, a209}, {t211, r209, f206, a210},
{t212, r202, f207, a204}, {t212, r201, f202, a201}, {t212, r205, f203, a208}};

Mund3 = {{t301, r301, f302, a308}, {t301, r307, f304, a304}, {t301, r309, f308, a308},
{t302, r305, f308, a305}, {t302, r306, f305, a311}, {t302, r310, f310, a305},
{t303, r307, f307, a309}, {t303, r308, f305, a301}, {t303, r303, f303, a307},
{t304, r302, f309, a304}, {t304, r303, f305, a310}, {t304, r304, f302, a309},
{t305, r308, f309, a312}, {t305, r302, f303, a306}, {t305, r301, f310, a304},
{t306, r309, f310, a308}, {t306, r310, f306, a303}, {t306, r302, f305, a312},
{t307, r307, f302, a301}, {t307, r308, f303, a311}, {t307, r305, f309, a305},
{t308, r306, f309, a309}, {t308, r309, f306, a306}, {t308, r310, f308, a310},
{t309, r301, f305, a303}, {t309, r304, f301, a302}, {t309, r306, f306, a303},
{t310, r303, f304, a306}, {t310, r305, f309, a307}, {t310, r308, f303, a309},
{t311, r310, f309, a310}, {t311, r305, f308, a301}, {t311, r301, f306, a305}};

Mund4 = {{t401, r409, f401, a405}, {t401, r407, f403, a403}, {t402, r410, f405, a406},
{t402, r408, f408, a404}, {t403, r407, f406, a405}, {t403, r404, f403, a403},
{t404, r411, f401, a407}, {t404, r401, f407, a408}, {t404, r409, f408, a406},
{t405, r402, f403, a408}, {t405, r408, f404, a406}, {t406, r405, f407, a407}, {t406, r411, f404, a408},
{t407, r407, f401, a402}, {t407, r409, f408, a406}, {t407, r404, f405, a405},
{t408, r402, f404, a403}, {t408, r410, f406, a408}, {t409, r406, f407, a407}, {t409, r403, f403, a401},
{t410, r402, f403, a402}, {t410, r406, f402, a406}, {t410, r410, f405, a402}, {t410, r405, f407, a402},
{t411, r409, f404, a408}, {t411, r403, f408, a403}, {t411, r408, f401, a407},
{t412, r407, f408, a407}, {t412, r404, f401, a404}, {t412, r406, f403, a403},
{t413, r401, f403, a403}, {t413, r411, f404, a402}, {t413, r402, f407, a404}};
```

# Fonctions

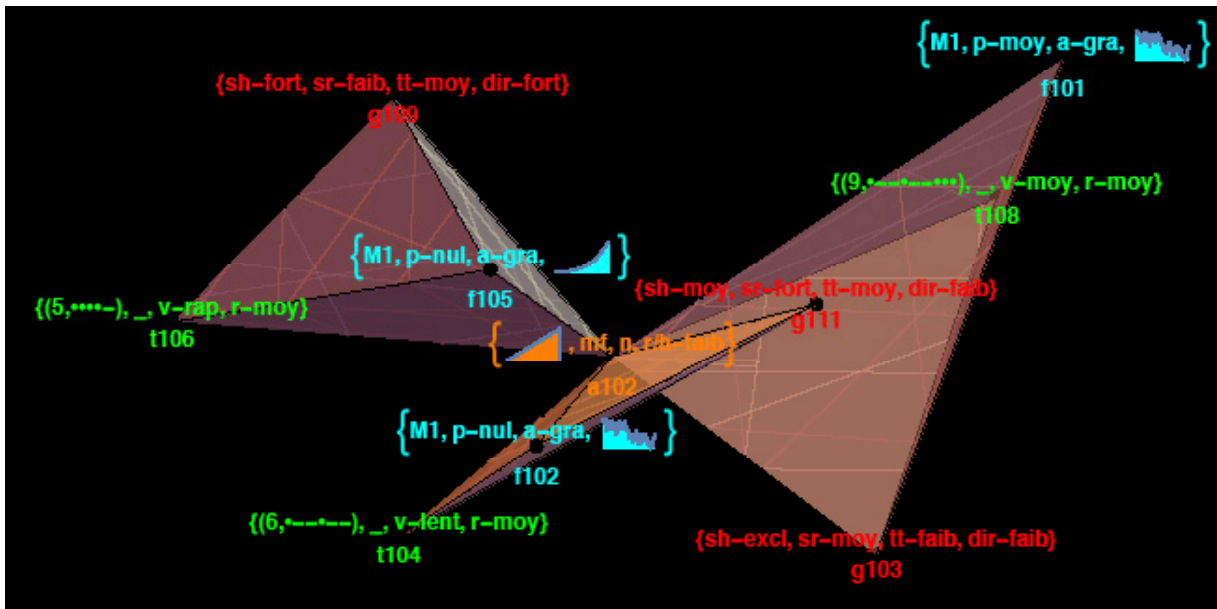
## Affichage d'un espace

```
CreateDocument[{TextCell[«Mund I», «Section»],
  Plonge3[Mund1, Mund1str, 17, Black, {t102, _, _, _}, {«t102», _, _, _}], Win-
  dowClickSelect -> True, WindowSize -> All] ;
```

## Filtrage d'un espace

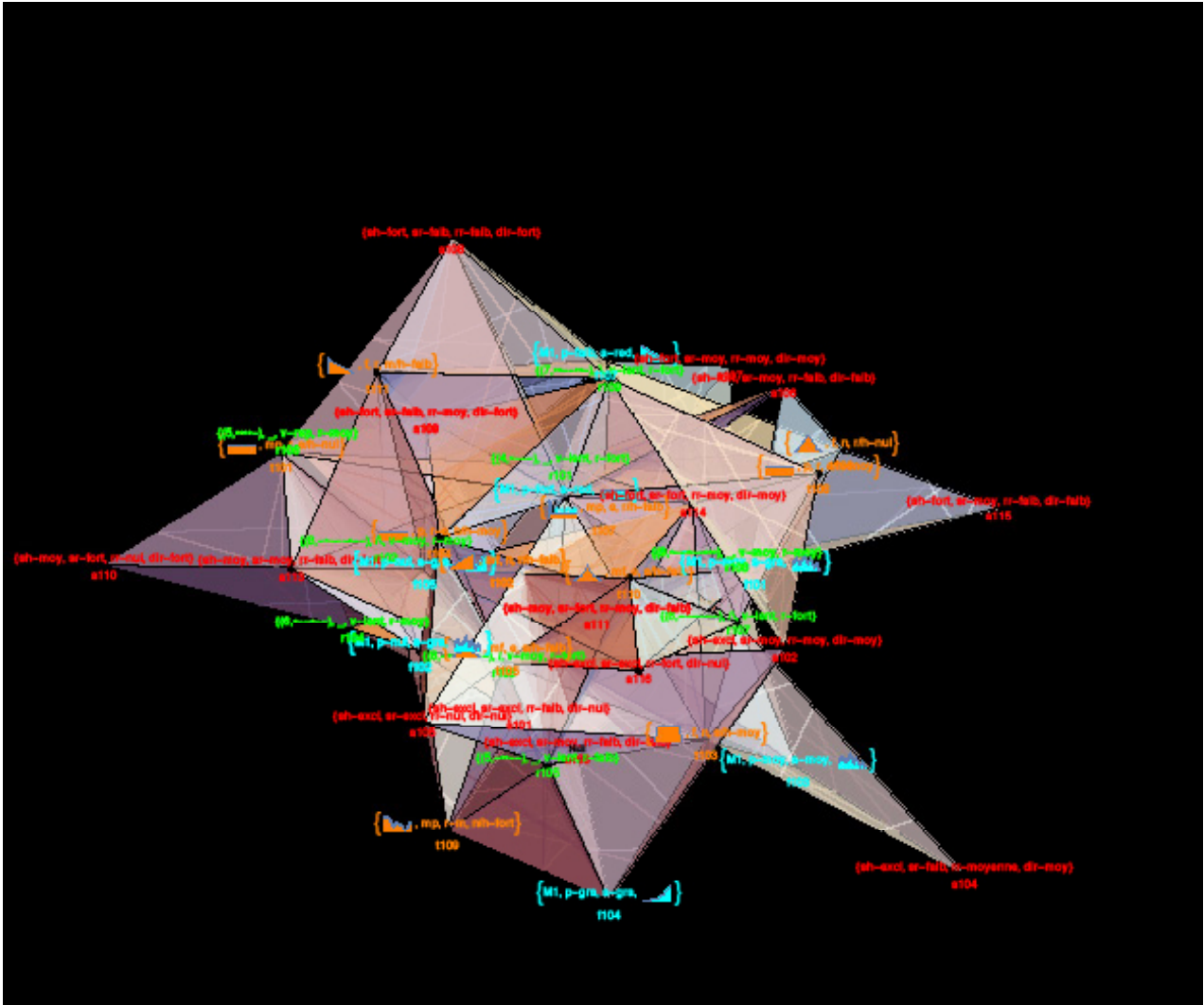
```
Filt[Mund1, Mund3str, 17, Black, {a102, _, _, _}, {«a102», _, _, _}]
```

Exemple d'un espace filtré, ici l'espace *Mund1* dans lequel on préserve uniquement les cellules ayant le simplexe a102 pour invariant.

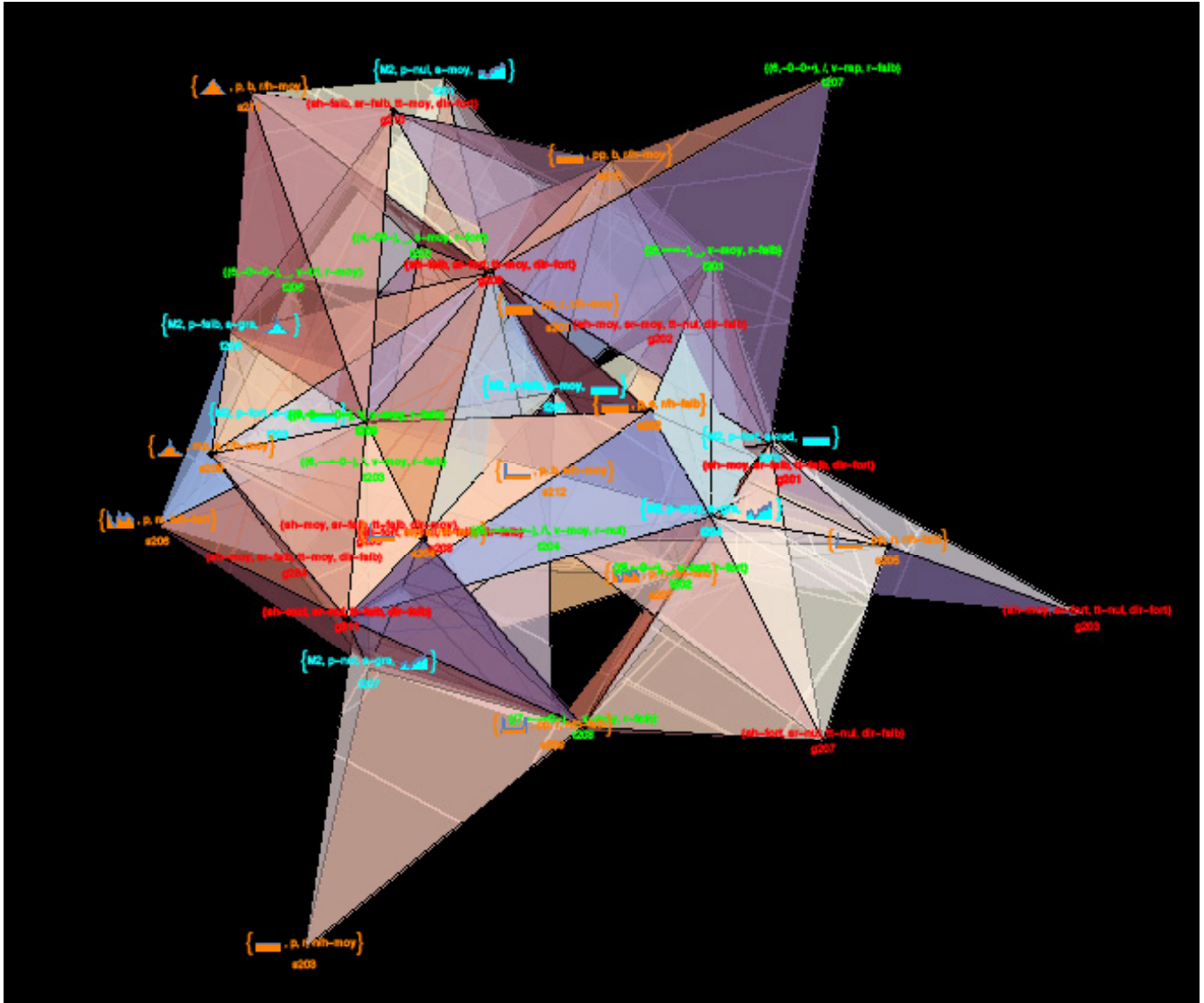


Les espaces suivants sont ceux qui ont été utilisés pour la composition de la pièce *Namenlosen*. Les espaces 2 et 3 ont également été utilisés pour la pièce *Sortir du noir*.

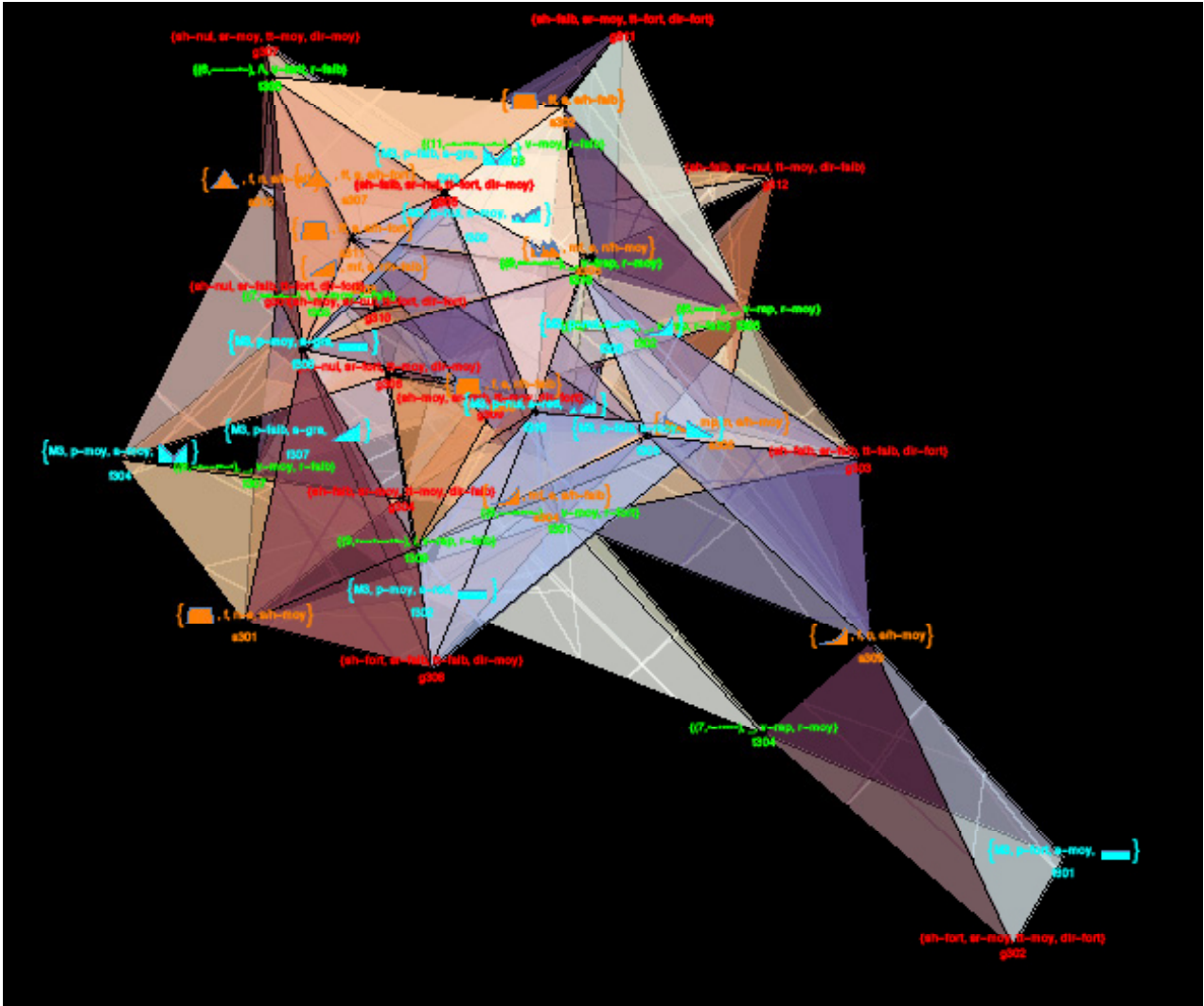
# Espace *Mund1*



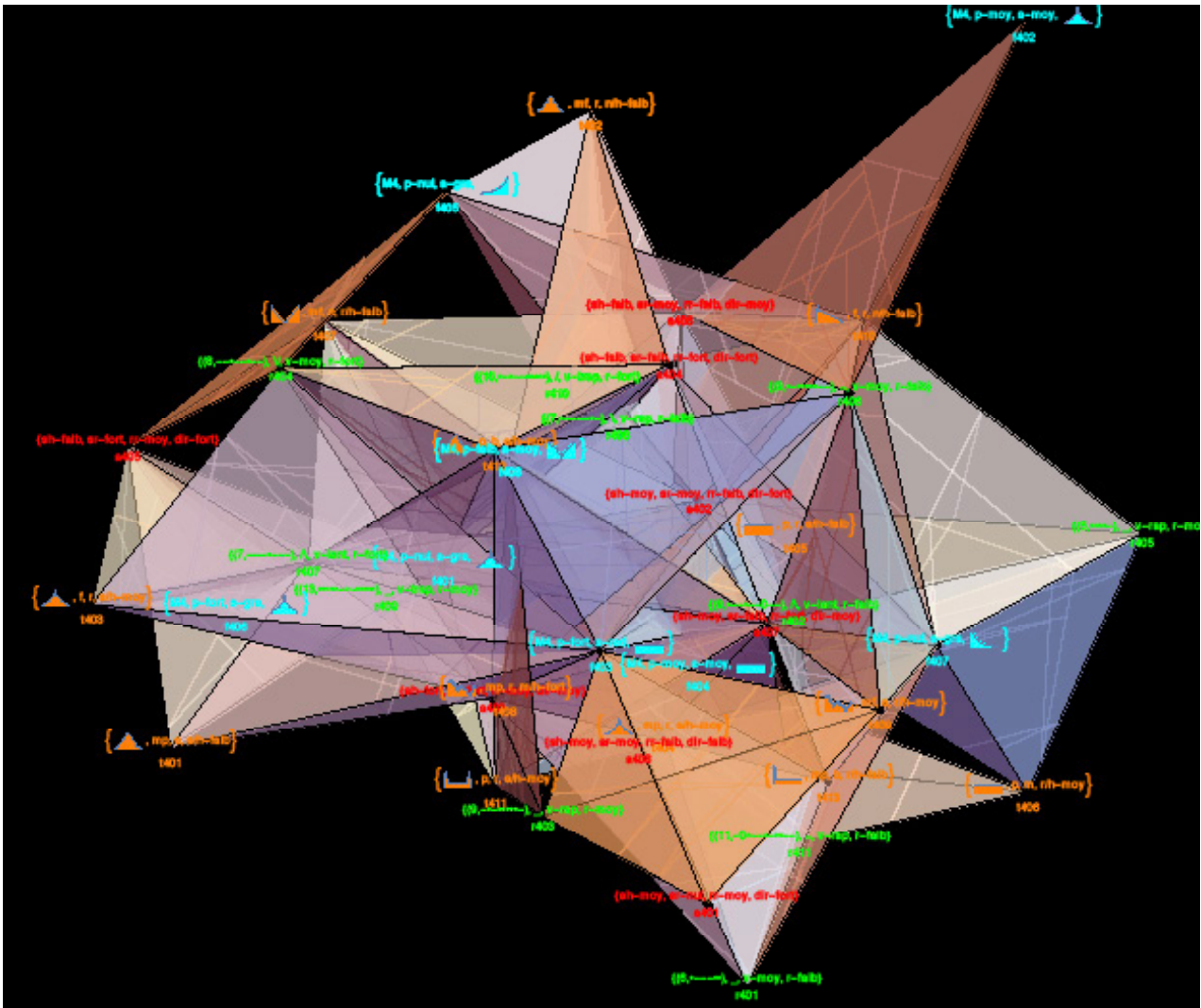
# Espace *Mund2*



# Espace *Mund3*



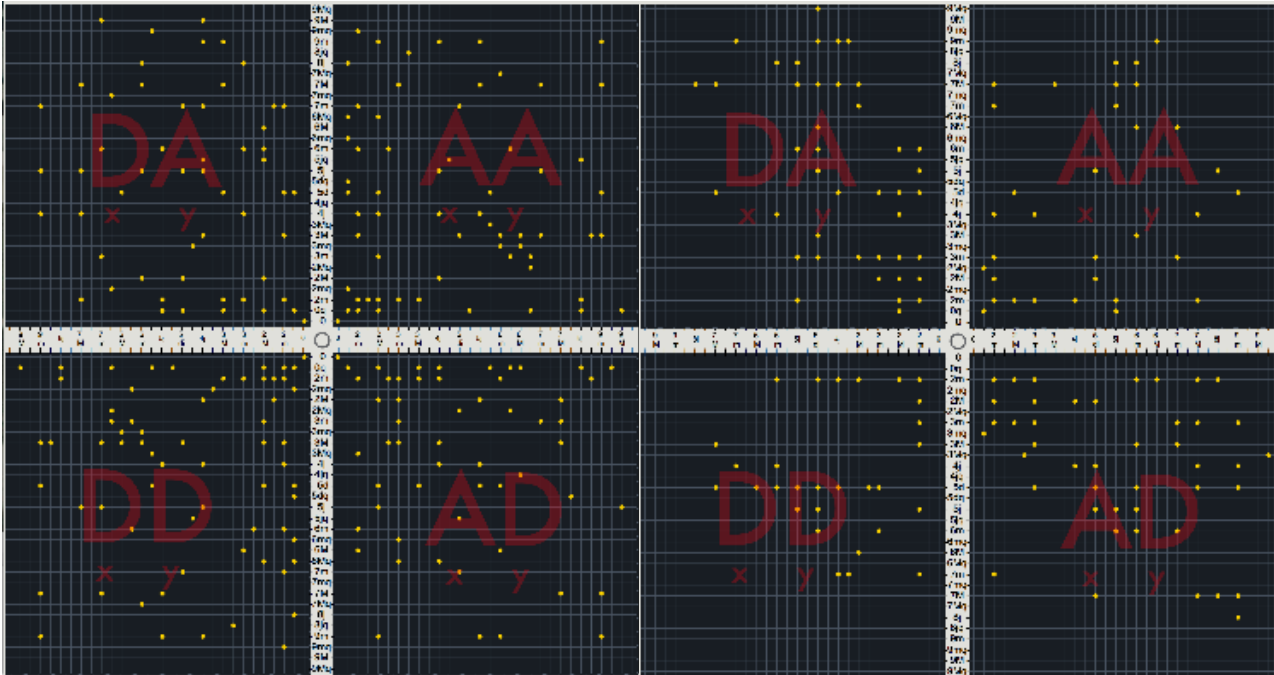
# Espace *Mund4*



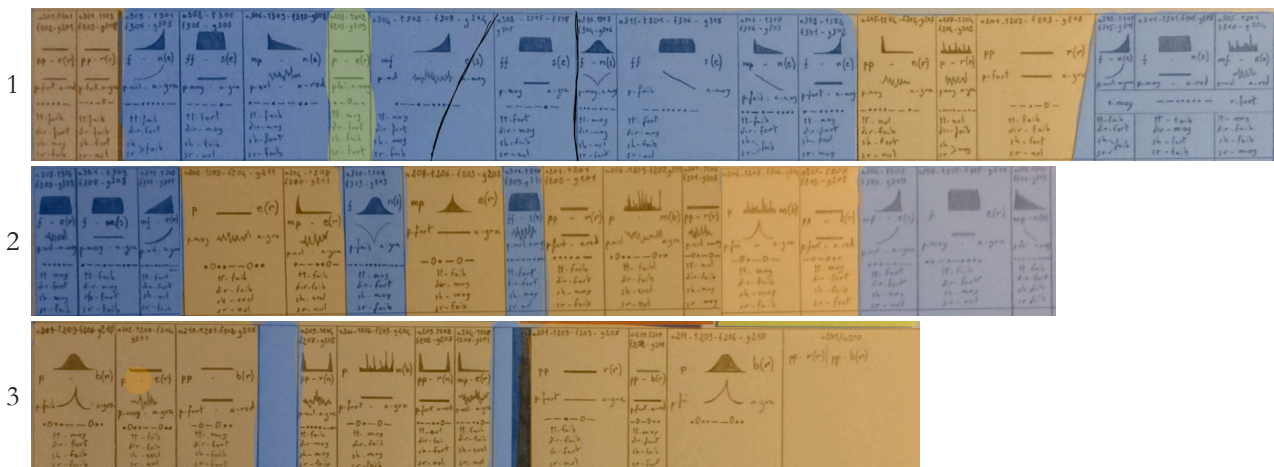
# SORTIR DU NOIR

Matrices d'intervalles utilisées pour *Sortir du noir*.

Les espaces compositionnels utilisés sont ceux de *Namenlosen* (Mund2 et Mund3).

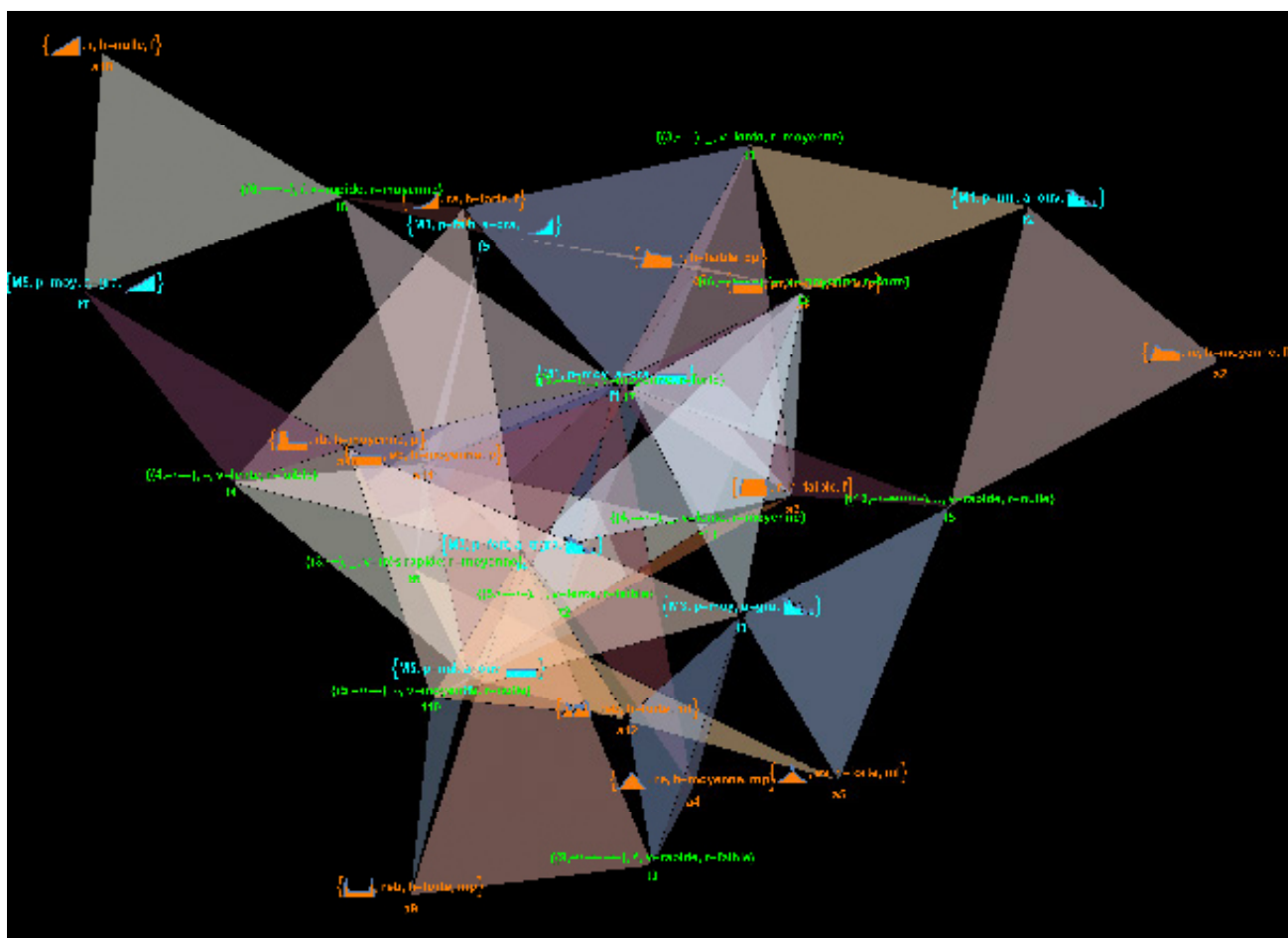


Le plan ci-dessous correspond aux différentes cellules utilisées pour la pièce. Les zones oranges correspondent aux cellules de l'espace Mund2 et les zones bleues concernent l'espace Mund3.



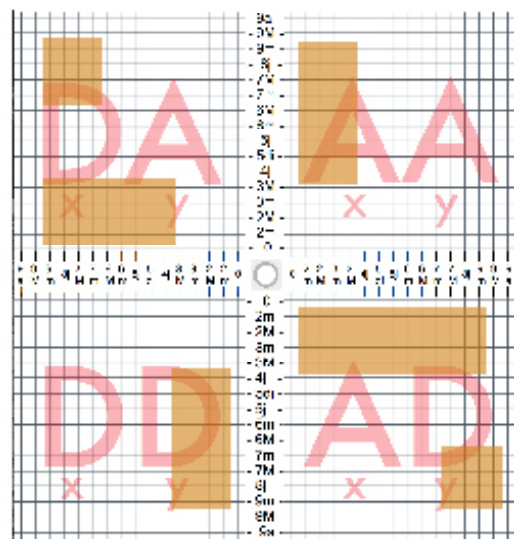
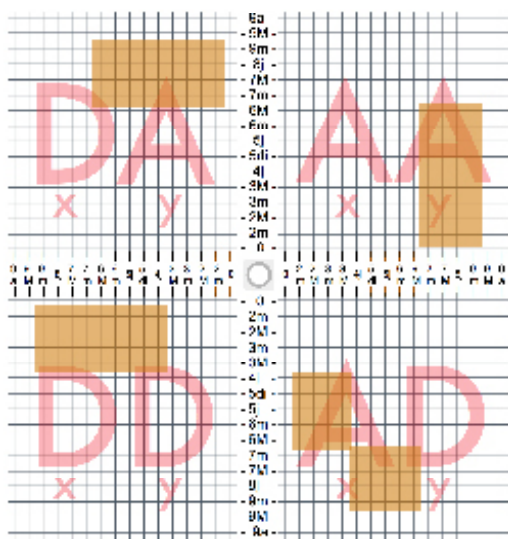
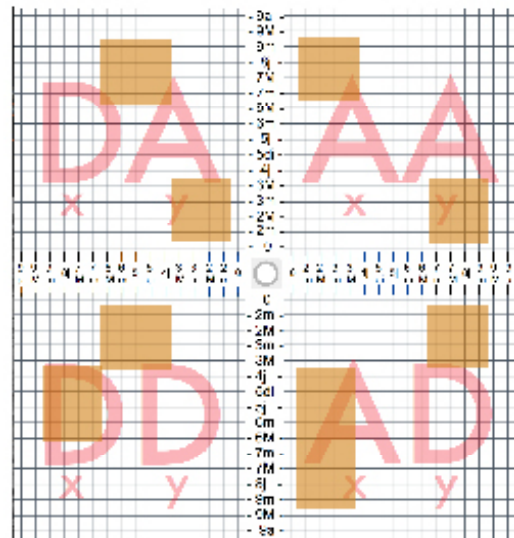
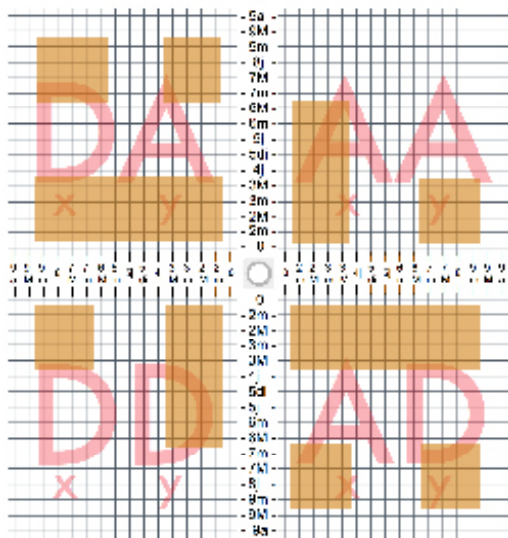
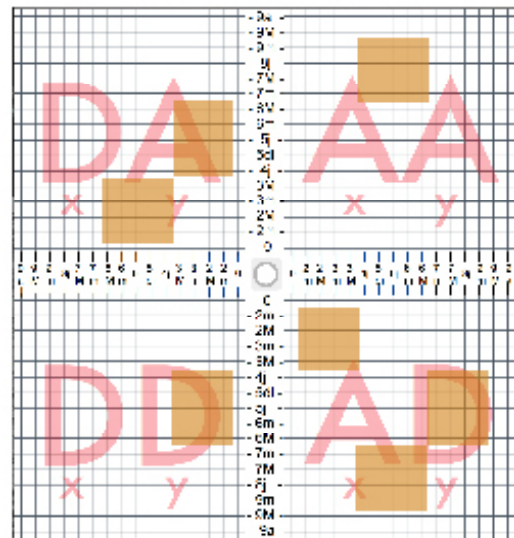
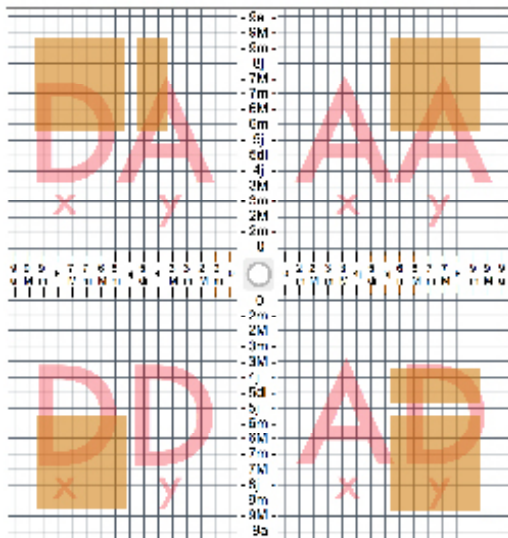
# THÉÂTRE D'OMBRES

Ci-dessous, l'un des espaces compositionnels ayant servi à la composition de la pièce. On peut noter que les espaces de cette pièce ne sont composés que de 2-simplexes. Les spécificités de nos espaces ont évolués à travers le temps et les différentes pièces composées.





Matrices d'intervalles



## ANNEXE 2



# ANTESCOFO : USER GUIDE

The Antescofo system couples machine listening and a specific programming language for compositional and performative purposes. It allows real-time synchronization of human musicians with computers during live performance.

## Interactive Music Systems

Mixed music (aka. interactive music) is the live association of acoustic instruments played by human musicians and electronic processes run on computers. Mixed music pieces feature real-time processes, as diverse as signal processing (sound effects, spatialization), signal synthesis, or message passing to multimedia software.

The specification of such processes and the definition of temporal constraints between musicians and electronics are critical issues in mixed music. They can be achieved through a program that connects music sheets and electronic processes. We call such a program an augmented score.

Indeed, a music score is a key tool for composers at authoring time and for musicians at performance time. Composers traditionally organize the musical events played by musicians on a virtual time line (expressed in beats). These objects share temporal relationships, such as structures of sequences (e.g., bars) or polyphony. To encompass all aspects of a mixed music piece, electronic actions have to share the same virtual time frame of the musical events, denoted in beats, and the same organization in hierarchical and sequential structures.

During live performance, musicians interpret the score with precise and personal timing, where the score time (in beats) is evaluated into the physical time (measurable in seconds). For the same score, different interpretations lead to different temporal deviations, and musician's actual tempo can vary drastically from the nominal tempo marks. This phenomenon depends on the individual performers and the interpretative context. To be executed in a musical way, electronic processes should follow the temporal deviations of the human performers.

## The Antescofo approach: coupling score following with a programming language

Achieving this goal starts by score following, a task defined as real-time automatic alignment of the performance (usually through its audio stream) on the music score. However, score following is only the first step toward musician-computer interaction; it enables such interactions but does not give any insight on the nature of the accompaniment and the way it is synchronized.

Antescofo is built on the strong coupling of machine listening and a specific programming language for compositional and performative purposes:

- The Listening module of Antescofo software infers the variability of the performance, through score following and tempo detection algorithms.

- And the Antescofo language
  - provides a generic expressive support for the design of complex musical scenarios between human musicians and computer mediums in real-time interactions
  - makes explicit the composer intentions on how computers and musicians are to perform together (for example should they play in a «call and response» manner, or should the musician takes the leads, etc.).

This way, the programmer/composer describes the interactive scenario with an augmented score, where musical objects stand next to computer programs, specifying temporal organizations for their live coordination. During each performance, human musicians “implement” the instrumental part of the score, while the system evaluates the electronic part taking into account the information provided by the listening module.

## Brief history of Antescofo

The Antescofo project started in 2007 as a joint project between a researcher (Arshia Cont) and a composer (Marco Stroppa) with the aim of composing an interactive piece for saxophone and live computer programs where the system acts as a Cyber Physical Music System. It rapidly became a system that couples a simple action language and machine listening.

The language was further used by other composers such as Jonathan Harvey, Philippe Manoury, Emmanuel Nunes and the system has been featured in world-class music concerts with ensembles such as the Los Angeles Philharmonic, New York Philharmonic, Berlin Philharmonic, BBC Orchestra and more.

In 2011, two computer scientists (Jean-Louis Giavitto from CNRS and Florent Jacquemard from Inria) joined the team and serious development on the language started with participation of José Echeveste (whose PhD was on Antescofo unique synchronization capabilities) and Philippe Cuvilier (whose PhD was on the use of temporal information in the listening machine). The new team MuTant was baptized during early 2012 as a joint venture between Ircam, CNRS, Inria and UPMC in Paris.

Antescofo has developed incrementally in line with user requests. The current language is highly dynamic and addresses requests from more than 40 serious artists using the system for their own artistic creation. Besides its incremental development with users and artists, the language is highly inspired by Synchronous Reactive languages such as ESTEREL and Cyber-Physical Systems.

## Structure of an Antescofo Score

An Antescofo score is a text file that is used for real-time score following (detecting the position and tempo of live musicians in a given score) and triggering electronics as written by the artists. Antescofo is thus used for computer arts involving live interaction and synchronisation between human and computerised actions.

An Antescofo score can be edited by any text editor. The users can find many syntax highlights on our Antescofo ForumUser. The package for the Sublime editor is particularly usefull, as coherence is preserved in real-time between the score loaded in Max and the score edited by Sublime.

There is also a dedicated GUI, AscoGraph, that can be used to edit an Antescofo score and to interact with a running Antescofo program. In this video, for example, we can see Antescofo playing with a musician and controlling both the electronic elements and the spatialization. The musician’s score (in a

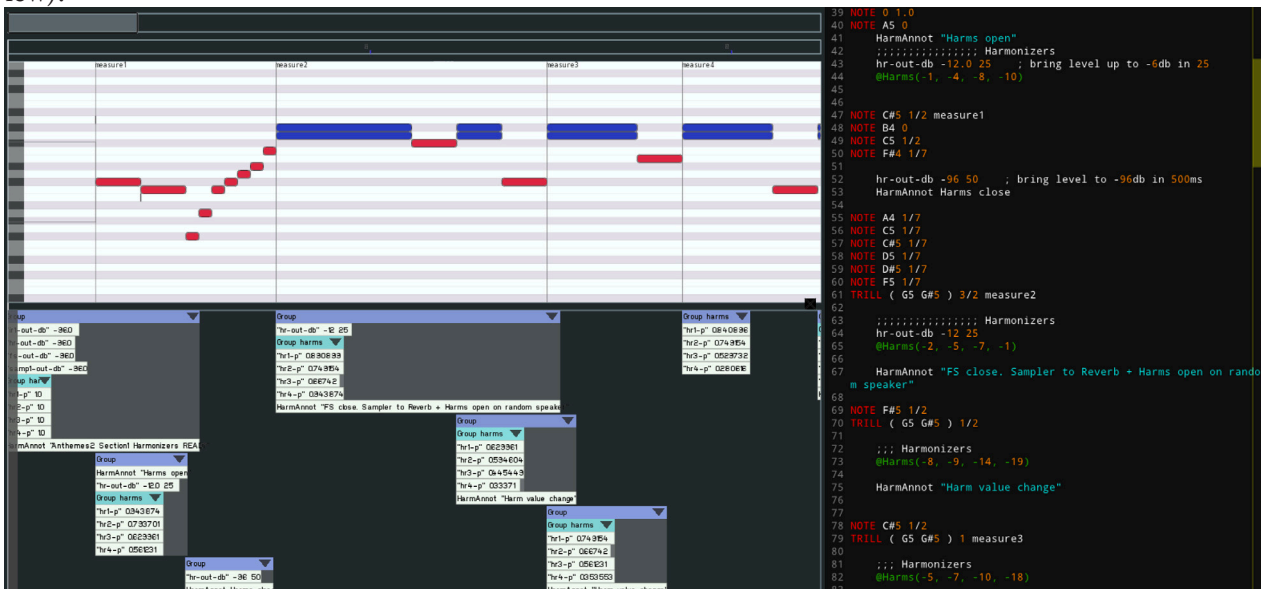
piano roll) and the electronic actions are both visible in the AscoGraph window.

## An interweaving of musical events and electronic actions

An Antescofo score describes both the human actions to be recognized and the machine's reactions to environmental input. A score thus has two main elements:

- EVENTS are elements to be recognized by the score follower or machine listener, describing the dynamics of the outside environment. They consist of NOTE, CHORD, TRILL and other elements discussed in details in section Event.
- ACTIONS are elements to be undertaken once corresponding event(s) or conditions have been recognized. Actions in Antescofo extend the good-old qlist object elements in MAX and PD with additional features which will be described in this document.

The figure below shows a simple example from the Composer Tutorial on Pierre Boulez' "Anthèmes 2" (1997) for violin and live electronics as seen in Ascograph (open the picture in a new tab for a larger view).



The left window shows a visual representation of Events and Actions, whereas the right segment shows the raw text score. Events in the score describe expected notes, trills and grace notes from the solo Violin, and Actions specify messages to be sent upon recognition of each event. In this example, we show-case actions for four real-time pitch shifter (or harmoniser), whose general volume is controlled by the `hr-out-db` parameter, and each shifter parameter separately controlled by `hr1-p` to `hr4-p`. The values for pitch shifters are in pitch-scale factor. Their corresponding musical value is described in the text score as comments (any text following a semi-colon ‘;’ is ignored in the score).

This score shows basic use of actions and events. Red text in the text-editor correspond to reserved keyword for Events. For details regarding available events and their corresponding syntax, see section Event. In this example, actions are basic message-passing to receivers in Max or Pd environments but with OSC, Antescofo can interact with many other softwares like CSound, Supercollider... Since they are isolated and discrete actions, we refer to them as Atomic Actions. As will be shown later, actions in Antescofo can use delays expressed in various time formats, and further include dynamic (i.e. real-time evaluated) expressions, data-structures and more. Details on action structures are discussed in section Actions.

## ANNEXE 2

The next figure shows a slightly more complex score corresponding to the beginning of “Tesla ou l’effet d’étrangeté” (2014) by composer Julia Blondeau for viola and live electronics as seen in Ascograph. The graphical representation on the left is a visual interpretation of the text score on the right. For easier viewing, messages are represented here by white circles (AscoGraph option).

The screenshot displays the Antescofo software interface. The top section shows a piano roll with notes and curves. The bottom section shows a detailed view of the score with various parameters and curves. The right side shows the text score with musical notation and Antescofo commands.

```

370 BPM 48
371 .NOTE 0 1
372 .----- mes 1 --- beat 1
373 NOTE G3 1/8 mes1
374 NOTE C#4 1/8
375 NOTE D3 1/8
376 NOTE B#3 1/8
377 TEMPO OFF
378 NOTE E6 8
379
380 GROUP synth @target (mes2)
381 {
382   SPAT2 xy 0 -1
383   ASCotoCS_SYNTH_L ("g1", 50.0, 8.0, 25.88, 0., 0.2, 5.2)
384   curve ampG1 @grain := 0.05s, @action := ASCotoCS_SYNTH_L c g1.amp $ampG1
385     {
386       { 0. } @type "cubic"
387       2 { 0.2 }
388       1 { 0.1 } @type "quad_out"
389       3 { 0.04 }
390     }
391   }
392   curve iterG1 @grain := 0.05s, @action := ASCotoCS_SYNTH_L c g1.templer $titerG1
393     { $titerG1
394       { 0. } @type "quad"
395       2 { 3. } @type "quad_in_out"
396       4 { 0. }
397     }
398   }
399   curve delmG1 @grain := 0.07s, @action := ASCotoCS_SYNTH_L c g1.multidel1 $delmG1
400     { $delmG1
401       { 0.2 }
402       3 { 18. }
403       5 { 0.2 }
404     }
405   }
406   curve freqG1 @grain := 0.08s, @action := ASCotoCS_SYNTH_L c g1.freq (@mid12hz($f
407     { $freqG1
408       { 88 } @type "quad"
409       19 { 87. }
410     }
411   }
412 }
413
414 3 GROUP orch
415 {
416   num_ech 4
417   PlayEch start
418   lvl_ech 0.3 500
419   curve spat9 @grain := 0.07s, @action := SPAT9 xy $x59 $y59
420     { $x59 $y59
421       { -1 -0.5 }
422       3/2 { 0 -1 }
423       2 { 1 -0.8 }
424     }
425   }
426   4 lvl_ech 0.3000
427 }
428
429 TEMPO ON
430 NOTE 0 0
431 NOTE Bb3 2/3
432 NOTE C#4 1
433 NOTE A5 3/2
434 TEMPO OFF
435 NOTE G#6 5
436
437 SPAT10 xv 0. 0.5
  
```

In the previous screenshot, the score for human musician contains many types of events (not all visible in text) with a mixture of discrete and continuous compound actions as written by the composer. At the moment you have to know that in Antescofo it exists 5 kinds of events :

NOTE : in red in the piano roll

CHORD : in green in the piano roll

TRILL for classical trill or tremolo : in blue in the piano roll

MULTI for glissandi (with one or more notes) : in yellow in the piano roll

EVENT for symbolic events

Tesla makes use of Compound Actions which consist of parallel groupings of Atomic Actions, as well as continuous actions introduced by keyword Curve. These elements correspond to different electronic actions. We can therefore compose the electronic part in parallel with the instrument part, together in an «augmented score». Effects, synthesis, spatialization and others can be controlled by Antescofo. Functions, Processes, Actors (objects) and Macros can be used to abstract and reuse these controls.

We will see that with Antescofo, we can compose with many different kind of time. Sequential time for the atomic actions, continuous time for curves, cyclic time for loop or «composed time» with processus that will see in due course.

## The file structure of an Antescofo augmented score

A textual score, or program, is written in a single file and loaded from there. The file itself can optionally include pointers to other score files, using the `@insert` feature:

```
@insert macro.asco.txt
@insert "file name with white space must be quoted"
```

The keyword can be capitalized: `@INSERT` as any other predefined `@`-identifier (i.e., identifiers that start with a `@` character). An included file may include other files. The `@insert` command is often used to store definitions and initialisation of the main Antescofo score. It will automatically create additional tabs in Ascograph text editor.

The `@insert_once` command is similar to `@insert` except that the file is included only once in the current score, when the directive is encountered the first time. The behavior makes possible to include a library of primitives in a set of files without the burden of taking care of their dependencies.

For the rest of this chapter, we will briefly introduce main elements in the language. Details will be left for dedicated chapters in the reference manual.

## Elements of an Antescofo Score

An Antescofo program is a sequence of events and actions. Events, recognized by the listening machine, are described in detail in chapter Events. Actions, outlined in chapter Actions are computations triggered upon the occurrence of an event or of another action. Actions can be dynamically parameterized by expressions and data structures, evaluated in real-time and described in detail in the Reference Manual.

Elements of the language can be categorized into six groups that correspond to various constructions permitted in the language:

- Comments: Any text starting by a semi-colon `;` or `//` is considered a comment and ignored by parser until the end of line (inline comment).
- Block (multi-line) C-Style comments starting with `/*` and ending with `*/` are also allowed.
- Keywords: are reserved words that introduce either Event or Action constructions. Examples include `Note` (for events) and `Group` (for compound actions).
- Simple identifiers: denote Max or PD receivers and are also used to specify the label of a musical event or of an action.
- `@`-identifiers: are words that start with `@` character. They either introduce a new definition or denote predefined functions, user-defined functions, user-defined macros, action attributes, or event attributes. The following `@`-identifiers are used to introduce new definitions: `@abort` `@broadcast` `@fun_def` `@init` `@macro_def` `@obj_def` `@pattern_def` `@proc_def` `@track_def` `@whenever`.
- `$`-identifiers: are words that start with `$` character. They correspond to user-defined variables or parameters in functions, processes, and object or macro definitions.
- `::`-identifiers: words starting with `::`, `obj::`, `pattern::` or `track::` refer respectively to processes, objects (actors), patterns or tracks.

User defined score elements including macros, processes and functions can only be employed after their definition in the score. We suggest putting them at the beginning of the file or to put them in a



separate file using the `@insert` command. They will be discussed in proceeding chapters.

## Simple identifiers: Antescofo keywords and reference to the host environment

The Antescofo language comes with a list Reserved Keywords for defining elementary score structures. Reserved keywords are case insensitive. They can be divided in two groups:

Event Keywords including `NOTE`, `CHORD`, `TRILL` and `MULTI` introduce musical events (see chapter Event in Antescofo) and are used to describe the music score to be recognised.

Action Keywords, such as `GROUP`, `LOOP` and more, specify computations that can be instantaneous (Atomic actions) or containers for other actions that have a duration (Compound actions).

Here is an example:

```
NOTE 60 1/2
  rcvr1 harm1 60 87 0.5
  rcvr2 ampSy 0.8 2
NOTE 62 1
  rcvr1 harm1 87 78 1.5
  1/2 print HELLO
```

In the example above, `rcvr1 harm1 60 87 0.5` and `rcvr2 ampSy 0.8 2` are actions that are hooked to event `NOTE 60 1/2`, and `1/2 print HELLO` denotes an action (sending to a receiver `print` in `max/pd`) with a delay of half-beat time. General syntax for atomic actions is described in chapter Atomic Actions. Event keywords can not be nested inside Action blocks. Event keywords are always defined at the top-level of the text score. Action keywords and blocks can be nested as will be discussed later.

The example shows another use of a simple identifiers: `rcvr1`, `rcvr2`, `harm1` `ampSy` and `print` are simple identifiers but are not reserved keywords. Here they refer to a receiver or a symbol in `Max/PD`. Simple identifiers that are not reserved keywords are case sensitive. In case a score requires the user to employ a reserved keyword inside a message, the user should wrap the keyword in quotes to avoid clash.

REMARK: An Antescofo text score is interpreted from top to bottom. In this sense, Event Sequence commands such as `BPM` or `variance` will affect lines that follow their appearance.

Example: The figure shows two simple scores. In the left score, the second tempo change to 90 BPM will be effective starting on the event with label `Measure2` and as a consequence, the delay `1/2` for its corresponding action is launched with 90 BPM. On the other hand, in the right score the tempo change will affect the chord following that event onwards and consequently, the action delay of `1/2` beat-time hooked on note `C5` corresponds to a score tempo of 60 BPM.

## ANNEXE 2

```
BPM 60
NOTE C4 1.0 Measure
CHORD (C4 E4) 2.0
NOTE G4 1.0
BPM 90
NOTE C5 1.0 Measure2
      1/2 print action1
CHORD (C5 E5) 2.0
NOTE A4 1.0
```

```
BPM 60
NOTE C4 1.0 Measure1
CHORD (C4 E4) 2.0
NOTE G4 1.0
NOTE C5 1.0 Measure2
      1/2 print action1
BPM 90
CHORD (C5 E5) 2.0
NOTE A4 1.0
```

## @-identifiers: Functions, Macros, and Attributes

A word beginning with a ‘@’ character is called a @-identifier. They have five purposes in Antecofo language:

1. in the processing of a file, some commands directly affect the parsing of this file: the @insert command is used to insert another file, and the commands @uid and @lid are used to generate on-the-fly fresh identifiers;
2. to introduce new definitions (functions, processes, tracks, patterns, etc.);
3. to specify various attributes of an event or an action;
4. to call internal functions that comes with Antecofo language as listed in chapter Library Functions;
5. and to call user-defined functions or macros.

Only ! ? . and \_ are allowed as special (non alphanumeric) characters after the @.

Note that in the first three cases, @-identifiers are reserved identifiers and thus are case insensitive, that is @tight, @TIGHT and @TIGHT are the same keyword. Reserved @-identifiers are listed here.

Users can define their own functions as shown in chapter Functions. These @-identifiers are case sensitive. Predefined functions are listed here.

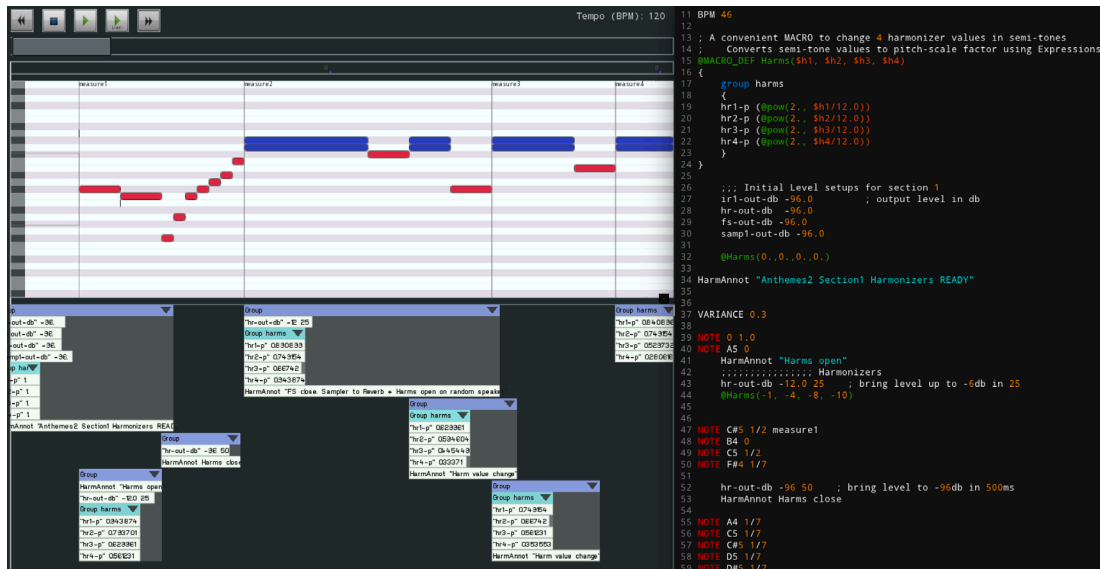
## \$-identifiers : Variables and Parameters

\$-identifiers like \$id or \$id\_1 are simple identifiers prefixed with a dollar sign. Only ! ? . and \_ are allowed as special characters after the \$. \$-identifier are used to give a name to variables (see section variables and as parameters for function, process and macro definition arguments. They are case-sensitive.

The figure below shows a rewrite of the excerpt of Pierre Boulez’ “Anthèmes 2” given here using a simple macro and employing basic @ and \$ identifiers. The harmoniser command is here defined as a macro for convenience and since it is being repeated through the same pattern. The content of the hr1-p to hr4-p actions inside the Macro use a mathematical expression using the internal function @pow to convert semi-tones to pitch-scale factor. As a result the Antescofo score is shorter and musically more readable. Variables passed to the macro definitions are \$-identifiers.

You can learn more on expressions and variables in chapter **expression** onwards.

## ANNEXE 2



The screenshot displays the Antescofo software interface. The top part shows a piano roll with a score for four measures. The bottom part shows a code editor with the following code:

```
11 BPM 46
12
13 ; A convenient MACRO to change 4 harmonizer values in semi-tones
14 ; Converts semi-tone values to pitch-scale factor using Expressions
15 #MACRO $p Harm($h1, $h2, $h3, $h4)
16 {
17   group harms
18   {
19     hr1-p ($pwm(2.. $h1/12.0))
20     hr2-p ($pwm(2.. $h2/12.0))
21     hr3-p ($pwm(2.. $h3/12.0))
22     hr4-p ($pwm(2.. $h4/12.0))
23   }
24 }
25
26 ;; Initial Level setups for section 1
27 ir1-out-db -96.0 ; output level in db
28 hr-out-db -96.0
29 fs-out-db -96.0
30 samp1-out-db -96.0
31
32 @harms(0..0..0..0)
33
34 HarmAnnot "Anthem2 Section1 Harmonizers READY"
35
36
37 VARIANCE 0.3
38
39 NOTE 0 1.0
40 NOTE A5 0
41 HarmAnnot "Harms open"
42 ::::::::::::::: Harmonizers
43 hr-out-db -12.0 25 ; bring level up to -6db in 25
44 @harms(1..4..8..10)
45
46
47 NOTE C#5 1/2 measure1
48 NOTE B4 0
49 NOTE C5 1/2
50 NOTE F#5 1/2
51
52 hr-out-db -96.50 ; bring level to -96db in 500ms
53 HarmAnnot Harms close
54
55 NOTE A4 1/7
56 NOTE C5 1/7
57 NOTE C#5 1/7
58 NOTE D5 1/7
59 NOTE D#5 1/7
```

## Events

An event in Antescofo terminology refers to elements that define what will probably happen outside your computers for real-time detection and recognition. In regular usage, they describe the music score to be played by the musician to follow. They are used by the listening machine to detect position and tempo of the musician (along other inferred parameters) which are by themselves used by the reactive and scheduling machine of Antescofo to produce synchronized accompaniments.

The listening machine is in charge of real-time automatic alignment of an audio stream played by one or more musicians, into a symbolic musical score described by Events. The Antescofo listening machine is polyphonic and constantly decodes the tempo of the live performer. This is achieved through explicit time models inspired by cognitive models of musical synchrony in the brain which provide both the tempo of the musician in real-time and also the anticipated position of future events (used for real-time scheduling).

This section describes Events and their syntax in Antescofo language. In a regular workflow, they can come from pre-composed music scores using MusicXML or MIDI import (see section score import). They can also be composed directly into the Antescofo text program.

## Event Specification

Events are detected by the listening machine in the audio stream. The specification of an event starts by a keyword defining the kind of event expected and some additional parameters:

```
NOTE pitch duration [label]
CHORD (pitch_list) duration [label]
TRILL ((pitch_list)*) duration [label]
MULTI ((pitch_list)*) duration [label]
MULTI (pitch_list) -> (pitch_list)) duration [label]
```

Here the '\*' is a metacharacter meaning "zero or more repetitions" of the preceding construction. Elements between square brackets '[' and ']' are optional but parentheses are literal elements that do ap-

pear in the code (they are not metacharacters).

TRILL and MULTI are examples of compound events organizing a set of NOTES in time. Thus they can accept one or several pitch\_lists. pitch\_lists in TRILL and MULTI are distinguished by their surrounding parentheses. See the next section for a more musically oriented explanation.

Events specification can be optionally followed by some attributes as discussed in the Event Attributes section below. Events must end by a carriage return. In other word, you are allowed to define only one event per line.

There is an additional kind of event :

**EVENT** d

also followed by a mandatory duration d, which correspond to a fake event triggered manually by the “nextevent” button on the graphical interface or by the “nextevent” message to the antescofo object in MAX/PD.

Parameters for event specification are described below.

### Pitch

pitch (used in NOTE) can take the following forms:

MIDI number (e.g. 69 and 70),

MIDI cent number (e.g. 6900 and 7000),

Standard Pitch Name (e.g. A4 and A#4).

For microtonal notations, one can use either MIDI cent (e.g. 6900) or Pitch Name standard and MIDI cent deviations using '+' or '-' (e.g. NOTE A4+50 and NOTE A#4+50 or NOTE B4-50).

```
CHORD (A4+50 A#4+50 B4-50 Bx4-50 C##4+50 C##4-50)
```

```
print OK
```

a minus sign - may precede the previous specification to specify that the current note is a continuation of a note with the same pitch in the preceding event:

```
CHORD (C4 D5) 1
```

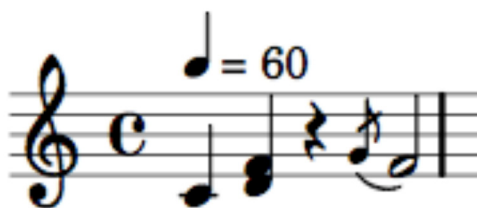
```
CHORD (-C4 D3) 1/2
```

### Pitch\_list

Pitch\_list is a set containing one or more pitches (used to define content of a CHORD). For example, the following line defines a C-Major chord composed of C4, E4, G4:

```
CHORD ( C4 64 6700 )
```

### Duration



Duration is a mandatory specification for all events. The of duration an event is specified in beats either by an integer (1), the ratio of two integers (4/3) or a float (1.0).

### Label

Optionally, users can define labels on events as a simple identifier or as a string, useful for browsing inside the score and for visualisation purposes. For example measure1 is an accepted label. If you intend to use space or mathematical symbols inside your string, you should surround them with quotations such as «measure 1» or «measure-1»


## Events as Containers

Each event keyword in Antescofo in the above listing can be seen as a container with specific behavior and given nominal durations. A NOTE is a container of one pitch. A CHORD contains a vector of pitches. The figure below shows an example including simple notes and chords written in Antescofo:

```

BPM 60
NOTE C4 1.0
CHORD (D4 F4) 1.0

```



```

NOTE 0 1.0 ; a silence
NOTE G4 0.0 ; a grace note with duration zero
NOTE F4 2.0

```

The two additional keywords TRILL and MULTI also define containers with specific extended behaviors:

## TRILL

Similar to trills in classical music, a TRILL is a container of events either as atomic pitches or chords, where the internal elements can happen in any specific order. Additionally, internal events in a TRILL are not obliged to happen in the environment. This way, can be additionally used to notate improvisation boxes where musicians are free to choose elements. A TRILL is considered as a global event with a nominal relative duration. Figure below shows basic examples for Trill.

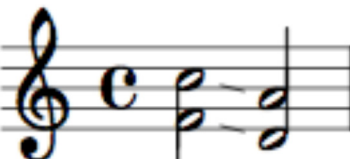
```

TRILL (A4 B4) 1.0
NOTE 0 1.0 ; a silence
TRILL ( (C5 E5) (D5 F5) ) 2.0

```


## MULTI

Similar to TRILL, a MULTI is a compound event (that can contain notes, chords or event trills) but where the order of actions are to be respected and decoded accordingly in the listening machine. They can model continuous events such as glissando. Additionally, a MULTI contents can be trills. To achieve this, it suffices to insert a character after the pitch\_list closure. The next example shows a glissandi between chords written by MULTI.



```
MULTI ( (F4 C5) -> (D4 A4) ) 4.0
```

*Compound Events*



Events can be combined and correspond to specific music notations. For example, a classical tremolo can be notated as a TRILL with one event (note or chord) inside. The next figure shows a glissando whose internal elements are tremolo. In this case, the prime ' next to each chord group indicate that the elements inside the MULTI are TRILL instead of regular notes or chords.



**MULTI** ( (C5 G5)' -> (D4 F4)' ) 2.0

The figure below shows a typical polyphonic situation on piano where the right-hand is playing a regular trill, and the left hand regular notes and chords. In this case, the score is to be segmented at each event onset as TRILL whose elements would become the trill element plus the static notes or chords in the left-hand.

```

TRILL ( (A4 A2) (B4 A2) )           1/2
TRILL ( (A4 D3) (B4 D3) )           1/2
TRILL ( (A4 C3 E3) (B4 C3 E3) )     1/2
TRILL ( (A4 D3) (B4 D3) )           1/2
TRILL ( A4 B4 )           2.0

```

### *Event Attributes*

Attributes in Antescofo are keywords following an @ character after the definition of the event. There are four kinds of event attributes and they are all optional.

- The keyword *fermata* (or @fermata) specifies that this event has a fermata signature. A Fermata event can last longer and arriving and leaving it does not contribute to the tempo decoding of the performance.
- The keyword *pizz* (or @pizz) specifies that the event is a string pizzicato. This usually helps Score Follower stability.
- The keyword *hook* (or @hook) specifies that this event cannot be missed (the listening machine needs to wait the occurrence of this event and cannot presume that it can be missed).
- The keyword *jump* (or @jump) is followed by a comma separated list of simple identifiers referring to the label of an event in the score. This attribute specifies that the event can be followed by several continuations: the next event in the score, as well as the events listed by the @jump.

These attribute can be given in any order. For instance:

```
Note D4 1 here @fermata @jump l1, l2
```

defines an event labeled by here which is potentially followed by the next event (in the file) or the

events labeled by l1 or l2 in the score. It has a fermata attribute. Note that

`Note D4 1 @jump l1, l2 here`

corresponds to the same specification: here is not interpreted as the argument of the jump but as a label for the event because there is no comma after l2.

### *Event Label*

A simple identifier or a string or an integer acts as a label for this event. There can be several such labels. If the label is a simple identifier, its \$-form can be used in a expression elsewhere in the score to denote the time in beat of the onset of the event.

### *The @modulate Attribute*

The @modulate attribute can be used on a BPM specification, not on an event. It specifies that the tempo must be modulated to the pro rata of the actual tempo of the performer. For example, if a BPM 60 is specified in the score, and the actual tempo of the performance is 70, then an indication of BPM 80 @modulate reset the tempo expected by the listening machine to  $80 \times 70 / 60 \approx 93.3$

## Actions in Brief

Think of actions as what Antescofo undertakes as a result of arriving at an instant in time. In traditional practices of interactive music, actions are message passing through qlist object in Max/Pd (or alternatively message boxes or COLL, PATTR objects in MAX). Actions in Antescofo allow more explicit organization of computer reactions over time and also with regards to themselves. See section message for a detailed description of the message passing mechanism.

Actions are divided into atomic actions performing an elementary computation or simple message passing, and compound actions. Compound actions group other actions allowing polyphony, loops and interpolated curves. An action is triggered by the event or the action that immediately precedes it.

In the new syntax, an action, either atomic or compound, starts with an optional delay, as defined hereafter. The old syntax for compound action, where the delay is after the keyword, is still recognized.

#### **Action Attributes**

Each action has some optional attributes which appear as a comma separated list:

```
atomic_action @att1, @att2 := value
compound_action @att1, @att2 := value { ... }
```

In this example, @att1 is an attribute limited to one keyword, and @att2 is an attribute that requires a parameter. The parameter is given after the optional sign :=.

Some attributes are specific to some kind of actions. There is however one attribute that can be specified for all actions: label. It is described in section Action Label. The attributes specific to a given kind of action are described in the section dedicated to this kind of action.

## Delays

An optional specification of a delay *d* can be given before any action *a*. This defines the amount of time between the previous event or the previous action in the score and the computation of *a*. At the expiration of the delay, we say that the action is fired (we use also the word triggered or launched). Thus, the following sequence

```
NOTE C3 2.0
    d1 action1
    d2 action2
NOTE D3 1.0
```

specifies that, in an ideal performance that adheres strictly to the temporal constraint specified in the score, *action1* will be fired *d1* after the recognition of the C note, and *action2* will be triggered *d2* after the launching of *action1*.

A delay can be any expression. This expression is evaluated when the preceding event is launched. That is, expression *d2* is evaluated in the logical instant where *action1* is computed. If the result is not a number, an error is signaled.

### Zero Delay

The absence of a delay is equivalent to a zero delay. A zero-delayed action is launched synchronously with the preceding action or with the recognition of its associated event. Synchronous actions are performed in the same logical instant and last zero time, cf. paragraph Logical Instant.

### Absolute and Relative Delay

A delay can be either absolute or relative. An absolute delay is expressed in seconds or milliseconds and refers to wall clock time or physical time. The qualifier *s* (respectively *ms*) is used to denote an absolute delay:

```
        a0
    1 s a1
(2*$v) ms a2
```

Action *a1* occurs one second after *a0* and *a2* occurs (2\*\$v) milliseconds after *a1*. If the qualifier *s* or *ms* is missing, the delay is expressed in beats and it is relative to the tempo of the enclosing group (see section local tempo).

### Evaluation of a Delay

In the previous example, the computed value of *a2*'s delay may depend of the date of the computation (for instance, the variable may be updated somewhere else in parallel). So, it is important to know when the computation of a delay occurs: it takes place when the previous action is launched, since the launching of this action is also the start of the delay. And the delay of the first action in a group is computed when the group is launched.

A second remark is that, once computed, the delay itself is not reevaluated until its expiration. However, the delay can be expressed in the relative tempo or relatively to a computed tempo and its mapping into the physical time is reevaluated as needed, that is, when the tempo changes.

### Synchronization Strategies

Delays can be seen as temporal relationships between actions. There are several ways, called synchronization strategies, to implement these temporal relationships at runtime. For instance, assuming that in



the first example of this section action2 actually occurs after the occurrence of NOTE D, one may count a delay of  $d1 + d2 - 2.0$  starting from NOTE D after launching action2. This approach will be for instance more tightly coupled with the stream of musical events. Synchronization strategies are discussed in section synchronization strategies.

## Label

Labels are used to refer to actions. Like events, actions can be labeled with

- a simple identifier,
- a string,
- an integer.

The labels of an action are specified using the `@name` keyword:

```
... @name := somelabel
... @name somelabel
```

One action can have several labels. Unlike with event labels, the `$`-identifier associated with an action label cannot be used to refer to the relative position of this action in the score<sup>1</sup>.

Compound actions have an optional identifier (section compound action). This is a simple identifier and acts as a label for the action without the burden to explicitly use the `@name` attribute.

## Action Execution

We write at the beginning of this chapter that actions are performed when arriving at an instant in time. But the specification of this date can take several forms. It can be

- the occurrence of a musical event;
- the occurrence of a logical event (see the whenever construction page and the pattern specification at page Whenever);
- the loading of the score (cf. the `@eval_when_load` construct at page Eval when Load);
- the signal spanned by an `@abort` action (see abort handler at page Abort);
- the sampling of a curve construct (page Curve);
- the instance of an iterative construct (pages Loop and ForAll);
- or the expiration of a delay starting with the triggering of another action.

## A Brief overview of Antescofo features

This section introduces some features that are very useful, especially for composers. If you are interested to learn more, take a look at the corresponding chapters in the Antescofo reference. Remember that everything having to do with control in Max or PD could certainly be replaced by an Antescofo Program. You will see that you can easily manage tabs, lists and some features that you need to control all the parameters of a concert patch.

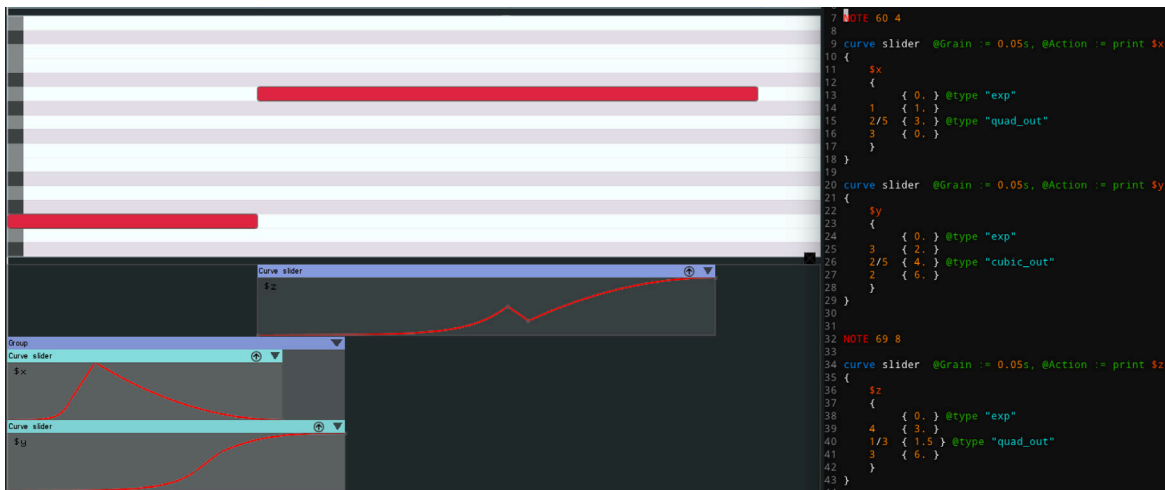
---

<sup>1</sup> There is no useful notion of position of an action in the score because the same action may be fired several times (actions inside a loop body or a or associated to a whenever or a process).

## A useful action : the curve

One of Antescofo's useful Keywords is the curve. Many composers use automations in their sequencers or use some «line» in Max. In Antescofo, you can write a large ensemble of curves (with many interpolation types) and create a score that controls all the effects, spatialization and synthesis with this curves. All the receivers in your Max or PD patch can receive the updated variable controlled by a curve.

The example below shows 3 simple curves with different types of interpolation. A receiver named «print» will receive the variables \$x, \$y and \$z.



## Make your life easier with macros !

Frequently, we use a same function that need some parameters in different moment of a piece. So you need to write many lines for a unique effect or synthesis. With a macro, you can denote that some parameters define a single musical entity. We think that is important to can choose between some writing style. Some different ways of thinking need different ways of writing !

The next example shows two ways to write a same thing.

; EXAMPLE 1 : the good old Q-list style !

```

NOTE 60 1
  SPAT_REV 0.2
  SPAT_X 1.
  SPAT_Y 0.8

  AddSynt_Hn 8
  AddSynt_F0 888
  AddSynt_Rev 1.4
  AddSynt_Mod 0.1

```

```

NOTE 69 2
  SPAT_REV 0.8
  SPAT_X 0.
  SPAT_Y 1.2

  AddSynt_Hn 6
  AddSynt_F0 1857

```

```
AddSynt_Rev 1.8
AddSynt_Mod 0.
```

NOTE 63 4

```
SPAT_REV 3.2
SPAT_X 2.
SPAT_Y 0.
```

```
AddSynt_Hn 14
AddSynt_F0 68.4
AddSynt_Rev 2.3
AddSynt_Mod 0.002
```

; EXAMPLE 2 : In another way....

```
@macro_def SPAT ($REV, $X, $Y)
{
    SPAT_REV $REV
    SPAT_X $X
    SPAT_Y $Y
}
@macro_def AddSynt ($Hn, $F0, $Rev, $Mod)
{
    AddSynt_Hn $Hn
    AddSynt_F0 $F0
    AddSynt_Rev $Rev
    AddSynt_Mod $Mod
}
```

NOTE 60 1

```
@SPAT(0.2, 1., 0.8)
@AddSynt(8, 888, 1.4, 0.1)
```

NOTE 69 2

```
@SPAT(0.8, 0., 1.2)
@AddSynt(6, 1857, 1.8, 0.)
```

NOTE 63 4

```
@SPAT(3.2, 2., 0.)
@AddSynt(14, 68.4, 2.3, 0.002)
```

## Tour the loop

Sometimes, the situation calls for a loop (ask to Steve Reich!). You can use many loops and imbricated loops in Antescofo. For example, you can include a curve in a loop and dynamically modify it while advancing the the loop time. There are many ways to end a loop. Guess what will happen in the following examples :

```
loop ForEver 1
  { print «Try again!» } ; an «infinite» loop...

3.5 abort ForEver ; ... that you have the power to finish !
  print «That's enough!»

;-----
```

```

$cpt := 0
loop L 1.5
{
    $cpt := $cpt + 1
    0.5 print a1
    0.5 print a2
} until ($cpt >= 3) ; A conditional end

;-----
; the same with an another type of abort
loop L 1.5
{
    print a0
    0.5 print a1
    0.5 print a2
} during [4.5] ; A temporal constraint

```

## Build your own world

In Antescofo, you can create your own functions (see `@fun_def`) if you frequently need to carry out the same task (to set a diapason, for example). There are many features that help create musical entities and facilitate electronic score writing.

### *Why do you need data structures...*

Data structures come in handy in many situations. Some effects and syntheses involve a long list of parameters. In the Data Structures chapter, you will see many ways to create and manipulate different kinds of lists and data structures (see `map` and `tab`).

In the classical music notation, there are many symbols that each denote an ensemble of parameters. These symbols permit the musician to focus on the music and not on the parameters. You can have the same approach in Antescofo if you use macros, processes and data structures. For example, if you use a physical model for synthesis, it's very laborious to enumerate all the parameters in your score. In this case, you can use an ensemble of tabs as a “playing mode” library. So in your score, you will just have to write the name of the tab and not the ensemble of parameters. Like when you write *Sul ponticello* you don't have to describe to the musicians how to play that !

The figure below shows a short library for a string physical model and a process that permits interpolation between two “playing modes”. Don't worry about understanding all the syntax (that's what the reference manual is for) but remember that it's possible !

```

; Declaration des "modes de jeu" pour interpolation
@global $SulPont,$Ord,$SulTasto,$SulPontVib1,$SulPontVib2,$SulTastoVib1,$SulTastoVib2,$OrdVib1,$OrdVib2

; pression : 0.2 - 5
; position : 0.028 - 0.22

$SulPont := [3.20, 0.032, 3.6, 0.] ; pression, position, freqVib, vibamp
$SulPontVib1 := [3.20, 0.032, 1.3, 0.002]
$SulPontVib2 := [3.20, 0.032, 7.1, 0.003]
$Ord := [2.5, 0.12, 1.3, 0.]
$OrdVib1 := [2.5, 0.12, 1.3, 0.002]
$OrdVib2 := [2.5, 0.12, 7.1, 0.003]
$SulTasto := [4.23,0.2, 1.3, 0. ]
$SulTastoVib1 := [0.9,0.22, 1.3, 0.002]
$SulTastoVib2 := [0.9,0.22, 7.1, 0.003]

; Numero du GenBow, [ModeJeu1,dur1,type1,ModeJeu2, ... , durn,typen,ModeJeu], Grain
; ex ::MdJ_bow("bow1",[$SulPont,1,"cubic",$OrdVib2,2,"linear",$SulTasto] ,0.2)
@proc_def ::MdJ_bow($numBow,$TabMdJ,$grain)
{
  if (@size($TabMdJ)>=4)
  {
    ::M_dJ_bow1($numBow,$TabMdJ[0],$TabMdJ[3],$TabMdJ[1],$TabMdJ[2],$grain)
    ($TabMdJ[1]) ::MdJ_bow($numBow,(@cdr(@cdr(@cdr($TabMdJ)))),$grain)
  }
}

@proc_def ::M_dJ_bow1($numBow,$MdJ1,$MdJ2,$durInterp,$type,$grain)
{
  @local $interpolMdJB, $x
  ; creation de la courbe entre les valeurs d'interpolations de chaque valeur des 2 tableaux
  $interpolMdJB := [ NIM{ 0 ($MdJ1[$i]), 1 ($MdJ2[$i]) EXPR{ $type } } | $i in @size($MdJ1) ]
  ; "tete de lecture" de la courbe
  curve interp @grain := $grain s,
  @action := { ASC0toCS_SYNTH_L c ($numBow+".kpres") ([ $nim($x) | $nim in $
    interpolMdJB][0] ) ; prend la 1ere valeur du tableau...
    ASC0toCS_SYNTH_L c ($numBow+".kpos") ([ $nim($x) | $nim in $
    interpolMdJB][1] )
    ASC0toCS_SYNTH_L c ($numBow+".kvibf") ([ $nim($x) | $nim in $
    interpolMdJB][2] )
    ASC0toCS_SYNTH_L c ($numBow+".kvibamp") ([ $nim($x) | $nim in $
    interpolMdJB][3] ) }
  { $x
    {
      $durInterp { 1. }
    }
  }
}
}

```

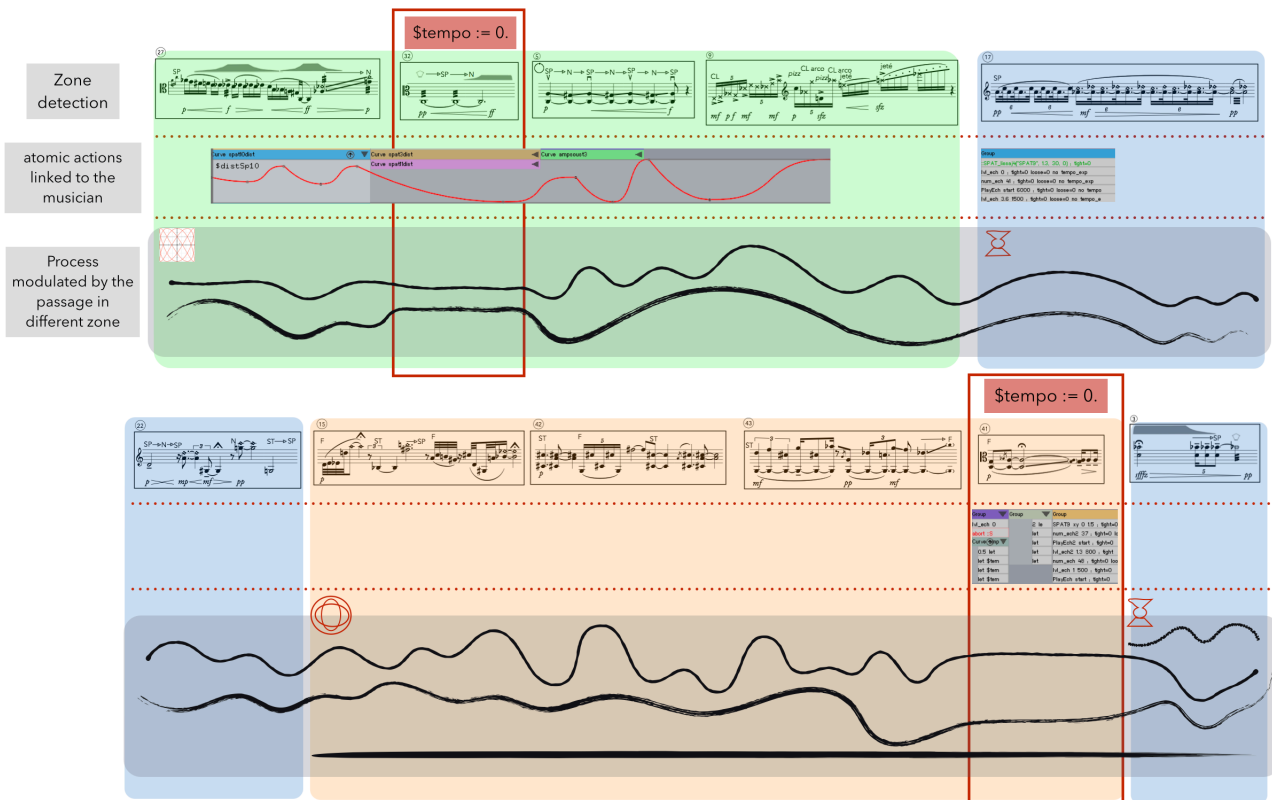
### *In processes we trust*

If you have to create complex processes that can heard some extra-parameters of the score (like audio descriptors, patterns....), Antescofo provides some dynamic features like the whenever or the processus that permit to write a real musical entity with some musical evolutions.

This process are different, in the way of thinking, of the classical score. The process is a kind of «daemon» that can be launched when it's needed and that can be aborted at the good moment. The process move in parallel with the score but take account of the musician's tempo (see \$RT\_TEMPO).

In the figure below, a musician can choose a path in an ensemble of short extracts. Different zones are associated at this extracts. In the score, some atomic actions are classically played with the score of musician, in a sequential way. In parralel, a process is launched and evolve depending on area where the musician is. The process can be seeing as an entity that evolve both with the musician and its own independant evolutions.

## ANNEXE 2



©Julia Blondeau - 2014

Create your own process, macro and function library that you use in all of your pieces.

### *A conditional world*

Sometimes we need to specify conditional actions. In Antescofo, the constructions `if` and `switch` are made for that. A conditional action is a compound action that performs different actions depending on whether a programmer-specified boolean condition evaluates to true or false.

You want launch a group of actions only if the musician plays at a particular amplitude ? You have to use this kind of code :

```

if ($musAmp >= 1.2)
{
    synt_receiver bang
}
else
{
    print «Hey! You're playing too softly!»
}

```

But, note that this kind of `if` is evaluated when it is launched. So... it is useful but you maybe have to watch at variable during all the time of the performance. In this context, you need a dynamic construction that look permanently the value of your variable. You need the `whenever` construction ! In the same idea of before, if you want to know when your musician is playing too loud, you can write something like this :

```
whenever ($musAmp >= 1.2)
{
    synt_receiver bang
}
```

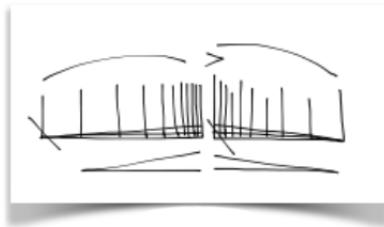
When the `whenever` statement is launched, the variable that it's given is permanently monitored and you will always know when the `whenever`'s condition is true.

It's as easy as pie!

### *Become the time master*

In Antescofo, all the electronic actions are launched in the musician's time. The internal variable `$RT_TEMPO` give the tempo of the musician in real time. So, when you write your score, you can be sure that it will be synchronize with the musician (if you interested by the synchronization question, take a look at the chapter Synchronization Strategies).

It's great ! But..., perhaps you would like to impose your time ! And, maybe, you had written an electronic phrase that sound too steep and you would like to introduce more softness in the "electronic phrasing". You can want to write an *accelerando*, but write the absolute time values is so laborious...



So, in Antescofo language, all the `group`, `process`, `loop`, `curve` ... can be "time controlled". This means that for each group, you can impose a tempo (BPM) or better: an evolution of tempo !

The attribute `@tempo` is made for that. If you see the previous example, you can see that for two periods of time, the process is like "time frozen" and you can see `::antescofo $tempo := 0` that is a simple example where you can stop for a moment any instance if you put its tempo at 0.

In the next example, a group fusee composed with atomic actions `::ASCOToCS_points` ... is time controlled by the variable `$tempfusee`. This variable is modified in time by a curve that give a phrasing to the group.

```

GROUP tempfus @tempo := 80
{
  curve tempfusee @grain := 0.1s
  { $tempfusee
    {
      4      { 40 } @type "quart"
      1      { 130 } @type "quad"
      3/4    { 50 }
      3/4    { 80 }
    }
  }
}

GROUP fusee @tempo := $tempfusee
{
  ::SPAT_lissaj2inv("SPAT7",1.4,6,0)
  curve ampexplo @grain := 0.05s
  { $ampexplo
    {
      2      { 0.09 } @type "quad"
      1      { 0.15 } @type "quad_in_out"
      1/2    { 0.04 }
      1/2    { 0.17 }
      1/2    { 0.09 }
    }
  }
  ::ASC0toCS_points("i33",13/6,$ampexplo,0.9,76)

  11/6 ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,79)
  1/6  ::ASC0toCS_points("i33",1/8,$ampexplo,0.1,78)
      ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,78.5)
  1/6  ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,77)
  1/6  ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,80)
      ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,80.5)
  1/6  ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,86)
      ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,86.5)
  1/6  ::ASC0toCS_points("i33",1/10,$ampexplo,0.7,91)
      ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,91.75)
  1/8  ::ASC0toCS_points("i33",1/10,$ampexplo,0.7,90)
  1/8  ::ASC0toCS_points("i33",1/10,$ampexplo,0.7,89)
      ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,89.5)
  1/8  ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,88)
  1/8  ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,87)
      ::ASC0toCS_points("i33",1/8,$ampexplo,0.7,86.5)
  1/8  ::ASC0toCS_points("i11",1/8,$ampexplo,0.7,86)
  1/8  ::ASC0toCS_points("i11",1/8,$ampexplo,0.7,85)
}

```

## Antescofo Workflow

In this chapter, we will see how to work with Antescofo, from the edition of the score to the interactions with Max, PureData or other softwares through OSC.

### Editing the Score



The first step in the development of an Antescofo Application is probably the authoring of the augmented score. An augmented score is a text program and can be edited with any text editor (like Sublime, TextWrangler or Emacs but not with a text processor like Office/Word).

However, Antescofo comes with a companion application: AscoGraph. Ascograph is a graphical tool that can be used to edit and then to control a running instance of Antescofo through OSC messages.

Ascograph and Antescofo are two independent applications but the coupling between Ascograph and an Antescofo instance running in MAX appears transparent for the user: a double-click on the Antescofo object launches Ascograph, saving a file under Ascograph will reload the file under the Antescofo object, loading a file under the Antescofo object will open it under Ascograph, etc.

Ascograph is available in the same bundle as Antescofo on the IRCAM Forum. It is still far from being stable but can be extremely useful for authoring/visualizing complex scores and for monitoring your live performances. Your feedback is extremely welcome.

## Importing Scores to Antescofo (import of Midi files and of MusicXML files)

Very often, the musical events to follow have already been specified through a score editor (Finale, Sibelius ...), or exist as a MIDI score.

It is possible to automatically import MIDI or MusicXML scores to Antescofo format to spare the burden of rewriting the followed part. This feature is available by drag and dropping MIDI or MusicXML files into AscoGraph. For multiple instrument score, care should be taken to extract required part in a separate MIDI or MusicXML file. The result is an Antescofo score that can then be modified at will.

In the figure below you can see the Antescofo importer that appears when you drag your score in AscoGraph.

Users employing these features should pay attention to the following points:

### *Importing MIDI scores to Antescofo*

The major problem with MIDI format is the absence of grace notes, trills, and glissandi. Such events will be shown as a series of raw event elements in the score.

Another major issue with MIDI import is the fact that in most cases, timing of note-offs are not decoded correctly (based on where the MIDI is coming from). Bad offset timing creates additional event with linked pitches (negative notes) with short durations. To avoid this, we recommend users to quantize their MIDI files using available software. The Antescofo importer does not quantize durations.

### *Importing MusicXML scores to Antescofo*

MusicXML is now the standard inter-exchange score format file between various score editing and visualisation software. It includes high-level vocabulary for events such as trills, grace notes and glissandi which can be converted to equivalent events. However, decoding and encoding MusicXML is not necessarily unique for the same score created by different software!

The Antescofo MusicXML import is optimized for MusicXML exports from FINALE software. Before converting MusicXML score to Antescofo, users are invited to take into account the following points and to correct their score accordingly, especially for complex contemporary music scores:

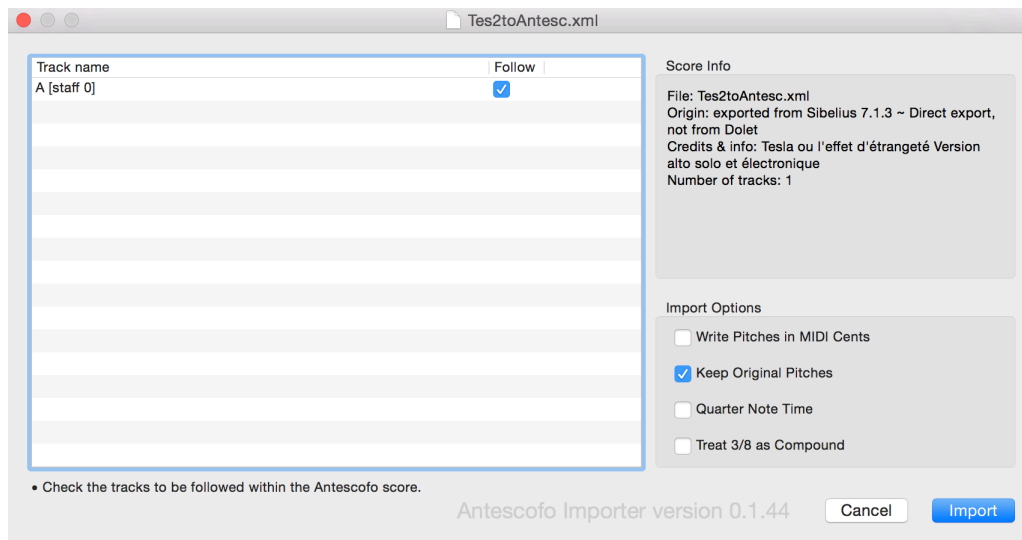
- Avoid using Layers: Merge all voices into one staff/voice before converting to MusicXML and

## ANNEXE 2

dragging to Ascograph. XML parsers sometimes generate errors and suppress some events when conflicts are detected between layers.

- Avoid using Graphical Elements in score editors. For example, Trills can only be translated to if they are non-graphical.
- If possible, avoid non-traditional note-heads in your editor to assure correct parsing for events.
- Avoid Hidden elements in your scores (used mostly to create beautiful layouts) as they can lead to unwanted results during conversion. Verify that durations in your score correspond to what you see and that they are not defined as hidden in the score.
- Verify your Trill elements after conversion as with some editors they can vary.

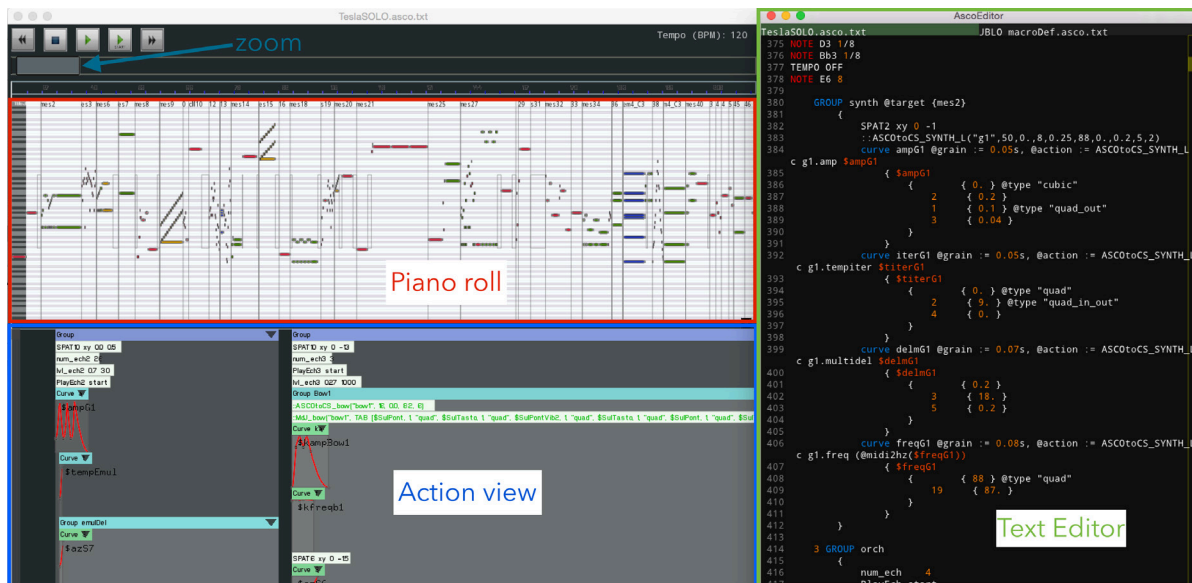
This feature is still experimental and we encourage users encountering problems to contact us through the Online User Group.



## Using AscoGraph

As you can see below (open the screenshot in another tab to have a larger view), the Ascograph interface contains 3 parts :

- the piano roll, where you can see the musician's score and the label at the top
- the action view, where you can see the graphical representation of the electronic score and where you can manually edit the curves.
- the text editor, where you can edit the musician's score and the electronic score.



At the top of the ascograph window, there are five buttons that correspond respectively to the Antescofo functions previous event, stop, play, start and next event that you can launch directly in your patch.

The play button simply sequences from the beginning of the score to the end, using given event timing and internal tempi, and undertaking actions wherever available.

The Play string function (Menu/Transport/Play string) only plays the selected part of your score. This enables to test on the fly Antescofo fragment (only actions are performed, musical events are ignored) and even allows a limited form of live-coding.

## The App Menu

The text Editor menu contains all the functions relevant to the text editor. Included is the display mode selector, where you can toggle between integrated, floating (default) and hidden. The View menu retains the functions pertaining to the visual editor.

## Color Scheme

AscoGraph comes with a new color scheme for both the text and visual editor. By default, we provide dark backgrounds for both. The philosophy behind this design choice is the fact that many of us use AscoGraph during live concerts and bright backgrounds create too much light on your screens which make things annoying for you and your audience who are most of the time in the dark.

Many users however prefer working on white background while typing. You can easily Import a new Color Scheme for AscoGraph's Text Editor from the Text editor Menu. You can design your own color scheme by doing an Export Color Scheme from the Text Editor menu, modify the XML file and import it back. Users interested in classical white background text editor can import Larry Nelson's Color Scheme in particular.

## Interaction Between Visual and Text Editors in AscoGraph

Clicking on a Musical Event or Action in the Visual Editor will bring the text editor to the corresponding text. Conversely, Right/Ctrl-click on an event in the text editor takes you to the corresponding place in the visual editor.

Vertical mouse-wheel (or double-finger on pad) gesture allows you to browse over the text-editor, as well as horizontal mouse-wheel (double-finger) gesture on the visual editor.

### *Shortcuts*

Holding CMD + mouse scroll: zoom in/out (visual & text editors) CMD + <number> switches the text editor between the open files (main .asco score and any @INSERT-ed files). You can also switch using the tabs at the top of the editor.

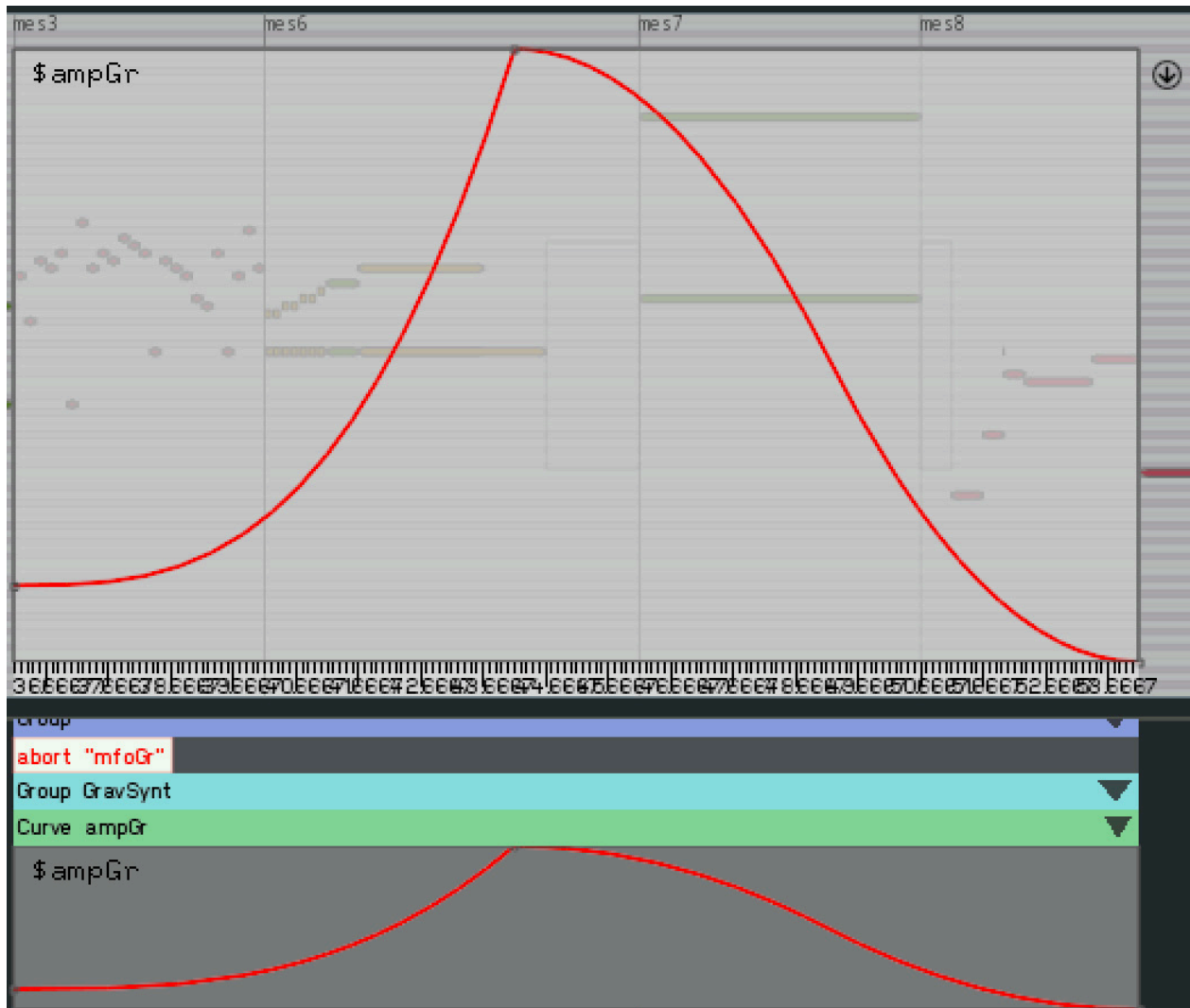
### *Edit your curves*

Ascograph is also very useful to visualize and edit the curves constructs.

You can Edit Antescofo Curves easily from the Visual Editor. Applying graphical changes will automatically create the corresponding text into the right place. See Nadir B's very useful YouTube Tutorials.

To start, you can create a curve after a note with the menu «create/actions/curve». After saving the score, if you see the action view, below the piano roll, you can see your curve.

If you click on the arrow at the right of the curve band, the curve is moved on the piano roll, to see the superpositions and you can add or move the curve's points. You can also choose the type of the interpolation between each points.



## Automatic Filewatch: Using Another Text Editor

Starting with AscoGraph version 0.25, there is an Automatic Filewatch integrated into the editor. This means that you can now use any of your favourite text editors instead of AscoGraph. The moment you save the Antescofo Score outside, it'll be automatically loaded (and visualized) in the AscoGraph window without any intervention. And the Antescofo object will reload the score automatically !

So you can use in parallel the Piano Roll and action view and still use your favorite text editor.

The addition of the Automatic Filewatch feature just makes score editing with AscoGraph more coherent with other editors as they all integrate automatic filewatch as well.

Max users can also use the Antescofo Max Object Autowatch attribute. This means that if the loaded score is modified elsewhere, it'll be automatically reloaded in the Max object. Obviously, you'd want to turn this attribute off during live concerts! Autowatch is currently not available for PureData objects but upcoming.

For the people that prefer use their own beloved text editor, we have created some syntax highlightings for Sublime Text, TextWrangler and... emacs ! You have just to download them in our Forum.

## Styling your score

Antescofo scores can be pretty big and it is important to grasp the elements at first glance. A syntax highlighter is included in AscoGraph and third party packages exist for Sublime and Atom, look at our Forum.

Antescofo scores can also be embedded in markdown (as illustrated by this documentation) using the Pygment highlighter or in latex using the lstlisting with a dedicated style.

The automatic indentation provided by editors like Sublime for Antescofo score is basic. It is always possible to send a `printfwd` message to the antescofo object. This command will write a file with an automatically indented version of the score together with various additional information (depending on the current verbosity). This file can be loaded again in Antescofo. Notice however that macros are expanded in this score and that the initial comments are not reported.

## Beyond score following...

Most people think Antescofo is exclusively used for mixed music and score following. That's partly true, but since the language exists and has been developed, Antescofo is also a great tool for electronic music.

### Antescofo as a sequencer

Speaking about time is very easy in Antescofo. AscoGraph and an Antescofo score can be seen like a sequencer, where all the actions are organised in time (if you click on the Play button, Antescofo becomes a sequencer) that you specify in the score.

You can write an electronic score with the beat notation and decide after to change the tempo. In this case you don't have to rewrite all the durations. Antescofo does the translation. This allows you to change the tempo (with BPM) in any place in your score.

As we have shown before, you can also work with different times in a same moment with the `@tempo` attribute and write complex polyrhythms.

With macros you can greatly simplify your score to focus on the musical questions instead of the technical realization.

## Hierarchical scores

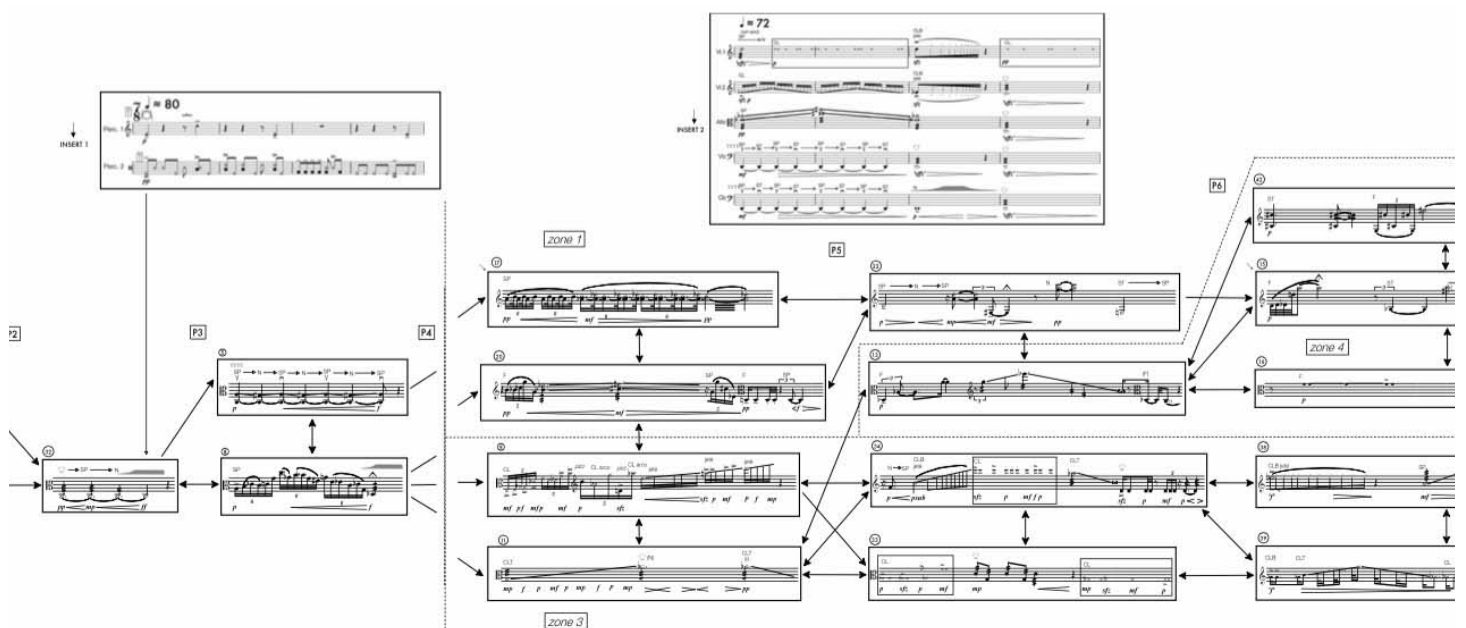
With the system of `GROUP`, you can write a hierarchical score where the different voices are denoted by different groups. You can also create tracks (see `@track_def`) to filter any kind of musical object or focus on a particular voice.

With the action view of AscoGraph you can easily see how the different voices of an electronic polyphony interact with each other.

## Open scores and installations

If you see the process and whenever constructions, you will see that these features can be easily used for installations or for open scores. You can use Antescofo without a “classical score”, without `NOTES`, `CHORD`... It means that you can program a reactive environment that “hears” sensors, audio descriptors...

If you compose open scores, you can take a look to the `@jump` keyword that is followed by a comma separated list of simple identifiers referring to the label of an event in the score. This attribute specifies that this event can be followed by several continuations: the next event in the score, as well as the events listed by the `@jump`. So you can compose a non-linear score with choice points leaved to the performer. You even can modify during the execution the list of allowed jumps, achieving a kind of “multi-graph” score.



## Beyond Max and Pure Data...

We have shown many uses of Antescofo in Max. However, with OSC, you can use Antescofo as controller of any kind of software that supports OSC. For example, SuperCollider and CSound are great Antescofo fellow players ! In the two next examples, Max is a kind of «pipe» between Antescofo, CSound or SuperCollider, IRCAM's Spat and audio descriptors. But Antescofo sends also many messages via OSC without Max.

### *A SuperCollider example*

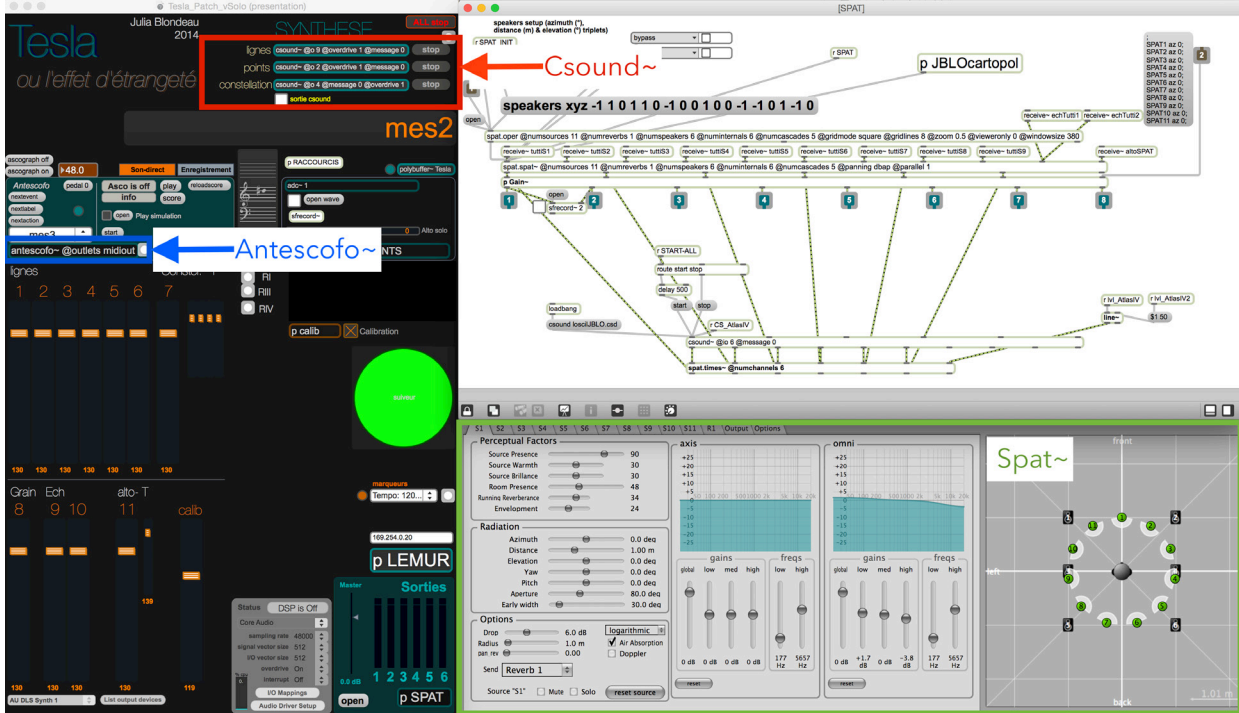
The figure below shows an environment created by José-Miguel Fernandez for his pieces where you can see 3 different softwares. In the background you can see the AscoGraph interface. At the left front you can see Max using Antescofo and descriptors (who are sent to whenever and process). At the right you can see a SuperCollider interface where processes are generated “on the fly” and displayed on the screen. In this example, Antescofo manages all the control parts and sends the parameters needed for synthesis and effects to SuperCollider. Fernandez uses many chaotic functions included in Antescofo and several of its data structures (e.g., map and tab).



## A CSound example

In the next example, Antescofo is used to send a score to CSound and to generate table for CSound. Here, Antescofo can replace the «.sco» file for CSound or the «score part» of the «.csd» in CSound. All the CSound synthesis is generated in real time and controlled by the object antescofo~.

The spatialization is controlled by Antescofo too with many curves and process that generate complex trajectory.



## OSC

Many people have been using Antescofo message passing strategies as defined above to interact with processes living outside MAX/PD (such as CSound, SuperCollider, etc.). To make their life easier, Antescofo comes with a built-in OSC host. The OSC protocol<sup>3</sup> can be used to interact with external processes using the UDP protocol. It can also be used to make two Antescofo objects interact within the same patch. Unlike MAX or PD messages, OSC messages can be sent and received at the level of the Antescofo program. The embedding of OSC in Antescofo is done through 4 primitives.

```
oscsend name host : port msg_prefix
```

```
oscrcv name port msg_prefix $v1 . . . $vn
```

see the OSC message chapter for more details.

## Be adventurous !

*« De toute manière, le compositeur, lors de ses voyages d'exploration, se voit à la fois comme Christophe Colomb et Don Quichotte – il débarque sur une terre inconnue et/ou tombe de cheval et atterrit de manière peu glorieuse, et à chaque fois en tout cas là où il ne pensait pas aboutir ; c'est ainsi seulement*

ANNEXE 2

*qu'il fait l'expérience de lui-même, qu'il se transforme, qu'il vient à lui-même. »<sup>2</sup>*

We have provided a brief overview of Antescofo, but a WORLD remains to be discovered... and experimented with! In the Reference Manual, you can more precisely research the previously addressed issues. You can decide to only take the examples given at the end of this documentation (snippets, How-to...) and copy them in your own score... or you can decide to roll up your sleeves and understand all the intricacies and secrets of Antescofo language to try to be an “aware Antescofo electronic composer !” ;-)

If you continue the adventure, you will see how to create a complex process or how to create a reactive environment where Antescofo can hear many things and react as you had composed! You will see how to manipulate time in Antescofo and how to synchronize your machine with a musician in real, musical way !

---

<sup>2</sup> Helmut Lachenmann, *Ecrits et entretiens*. Editions Contrechamps.  
173

ANNEXE 2

## ANNEXE 3



# Bibliothèque de spatialisation

```

; -----
; --- DICTIONNAIRES DE SPATIALISATION -----
; -- julia blondeau - 2017 -----

; -- FONCTIONS DE SPATIALISATION -----
; receivers concernés : to_matrix, to_hoa, to_vbap
;
; fonctions principales :
; - @current_assign_tab($spat_type) : donne l'état du tableau d'assignation des sources
de $spat_type
; - @set_all_spat_tempo($x) : change le tempo de l'ensemble des objets spat utilisés
;
; obj::spat
; -- $spat_son1 := obj::spat($sound_name,$duration,$source_of_sound_name,$spat_
type,$traj_type,$procArgs)
; -- _ := $spat_son1.set_synchro([4])
; -- _ := $spat_son1.set_tempo(180)
; -- _ := $spat_son1.current_sound_assign() : chemin du son au spat~ => CS_synth1 hoa 1
;
; Mouvements disponibles ($traj_type) :
; -- ::spat_traj_fun($sound_name,[$procSPAT,$dist,$speed,$offset,$elevation])
; -- ::spat_traj_pos($sound_name,[$a,$d,$e])
; -- ::spat_traj_ade($sound_name,[[ $a1,$d1,$e1,$a2,$d2,$e2,$interp1,$dur1],[ $a3,$d3,$e3,$
a4,$d4,$e4,$interp2,$dur2]])
; -- ::spat_traj_xye($sound_name,[[ $x1,$y1,$e1,$x2,$y2,$e2,$interp1,$dur1],[ $x3,$y3,$e3,$
x4,$y4,$e4,$interp2,$dur2]])
;
; Courbes de Lissajou
; -- ::lissaj1($SPAT,$sourceNum,$distanceL1,$vitesse1,$offset,$elevation)
; -- ::lissaj2($SPAT,$sourceNum,$distanceL1,$vitesse1,$offset,$elevation)
; -- ::lissaj3($SPAT,$sourceNum,$distanceL1,$vitesse1,$offset,$elevation)
; -- ::lissaj4($SPAT,$sourceNum,$distanceL1,$vitesse1,$offset,$elevation)
; -- ::lissaj5($SPAT,$sourceNum,$distanceL1,$vitesse1,$offset,$elevation)
; -- ::lissaj6($SPAT,$sourceNum,$distanceL1,$vitesse1,$offset,$elevation)
;
; Miscellenous
; -- ::spat_presence($sound_name,$presence)
;
; -----

$factDist := 1.

$target_spat_gen := 2 ; stratégie de synchro pour la spatialisation (horizon temporel)
$grain_spat := 0.08 ; grain des courbes d'envoi au spat (en secondes)

; tableaux d'affectation des sources automatique

$nombre_de_sources_hoa := 16
$nombre_de_sources_vbap := 16

; -----

$type_of_spat := map {(«vbap», [ 0 | ($nombre_de_sources_hoa)]), («hoa», [ 0 | ($nombre_de_
sources_hoa)])}

$assign_spat := map { } ; de type map { («sound_name», [«Csound1»,«hoa»,1]) }

; dictionnaire envoyant les données à la matrice (à rentrer à la main...)

```

## ANNEXE 3

```
$matrix_in := map { («csound1»,0), («csound2»,1), («csound3»,2), («csound4»,3), («c-
sound5»,4), («csound6»,5), («csound7»,6), («csound8»,7), («csound9»,8), («Player1»,9), («Pla-
yer2»,10), («Player3»,11), («Player4»,12), («Player5»,13), («Player6»,14), («ensemble1»,15), («
ensemble2»,16), («ensemble3»,17) } ;
```

```
$matrix_out := map { («hoa1»,0), («hoa2»,1), («hoa3»,2), («hoa4»,3), («hoa5»,4), («hoa6»,5), («
hoa7»,6), («hoa8»,7), («hoa9»,8), («hoa10»,9), («hoa11»,10), («hoa12»,11), («hoa13»,12), («hoa14
»,13), («hoa15»,14), («hoa16»,15), («vbap1»,16), («vbap2»,17), («vbap3»,18), («vbap4»,19), («vba
p5»,20), («vbap6»,21), («vbap7»,22), («vbap8»,23), («vbap9»,24), («vbap10»,25), («vbap11»,26), («
vbap12»,27), («vbap13»,28), («vbap14»,29), («vbap15»,30), («vbap16»,31), («scene1»,32), («scen
e2»,33), («scene3»,34), («scene4»,35) }
```

```
; ----- OBJET GENERAL SPAT -----
; objet-macro commun

@obj_def spat($sound_name,$duration,$source_of_sound_name,$spat_type,$traj_type,$procArgs)
{
  @local $obj_temp,$sync_strat,$temp_orig,$source_num

  @init { ; assignation d'une source
          $temp_orig := true
          $sync_strat := $target_spat_gen
          $obj_temp := $RT_TEMPO

          if (@size($type_of_spat($spat_type)) == @reduce(@+, $type_of_spat($spat_type)))
          {
            print "all sources of " $spat_type " are used !"
            print "the " $sound_name "'s spatialization have been aborted"
            abort $THISOBJ
          }
          else
          {
            ::spat_assign($sound_name,$source_of_sound_name,$spat_type)
            print ($type_of_spat($spat_type) $spat_type $sound_name

            $source_num := (($assign_spat($sound_name))[2])
            group spat_@tempo := ( $temp_orig ? $RT_TEMPO : $obj_temp ) @target [$sync_strat]
            {
              :: $traj_type($sound_name,$procArgs)

              ; abort final
              $duration let ($type_of_spat($spat_type)[$source_num-1]) := 0
              to_matrix ($matrix_in($source_of_sound_name)) ($matrix_out(($spat_type +
                $source_num))) 0
              $assign_spat := @remove($assign_spat,$sound_name)
              abort $THISOBJ ; ceci est un suicide !
            }
          }
        }

  @proc_def current_sound_assign()
  { print "chemin de " $sound_name " = " ($assign_spat($sound_name)[0]) " > " ($assign_spat($sound_name)[1]
    ) ($assign_spat($sound_name)[2]) }

  @method_def set_tempo($x)
  {
    $obj_temp := $x
    $temp_orig := false ; pour couper l arrivee de RT_TEMPO pour group spat
  }

  @method_def set_synchro($y) { $sync_strat := $y }

  @broadcast @set_all_spat_tempo($z) { $obj_temp := $x }
}

@macro_def @current_assign_tab($spat_type)
{ print "tableau d'assignation du " $spat_type " [ " ($type_of_spat($spat_type)) " ]" } ; état du tableau
d'assignations

; ----- Affectation d'une source sonore à une source du spat
; - cela évite de devoir se demander à quelle source on envoie le son
; - Ici la fonction ::spat_assign regarde quelles sont les sources libres et assigne
; - automatiquement une source du Spat au module sonore utilisé, puis envoie les infos
; nécessaires à la matrice

@fun_def @fun_assign_spat($tab,$index)
{
  if ($tab[0] == 0) { return ($index+1) }
  else
  { return @fun_assign_spat(@cdr($tab),($index+1)) }
}

@proc_def ::spat_assign($sound_name,$source_of_sound_name,$spat_type)
{
  @local $source_num
  $source_num := (@fun_assign_spat(($type_of_spat($spat_type)),0))
  $assign_spat := @add_pair($assign_spat,$sound_name,[$source_of_sound_name,$spat_type,$source_num])

  to_matrix ($matrix_in($source_of_sound_name)) ($matrix_out(($spat_type + $source_num))) 1
  let ($type_of_spat($spat_type)[$source_num-1]) := 1
} ; on ajoute une entrée au dictionnaire des assignations
; puis on envoie à la matrice les données correspondant à la clef de dico de matrice
; [CS_synth1,hoa_s1] si la source 1 est libre.
; on met enfin à jour vbap ou $hoa
; ::spat_assign("synth1mes34","CS_synth1","hoa")
```

## ANNEXE 3

```

; -----
; -- ::spat_traj_fun($sound_name, [$procSPAT, $dist, $speed, $offset, $elevation])
@proc_def ::spat_traj_fun($sound_name, $tab_args)
{
  @local $source, $spat_type, $procSPAT, $dist, $speed, $offset, $elevation

  ; arguments du proc
  $procSPAT := $tab_args[0]
  $dist := $tab_args[1]
  $speed := $tab_args[2]
  $offset := $tab_args[3]
  $elevation := $tab_args[4]

  $source := (($assign_spat($sound_name))[2])
  $spat_type := ("to_" + (($assign_spat($sound_name))[1]))
  :: $procSPAT($spat_type, $source, $dist, $speed, $offset, $elevation)
}

; -- ::spat_traj_pos($sound_name, [$a, $d, $e])
@proc_def ::spat_traj_pos($sound_name, $tab_args)
{
  @local $source, $spat_type, $a, $d, $e, $dd

  ; arguments du proc
  $a := $tab_args[0]
  $d := $tab_args[1]
  $e := $tab_args[2]

  $source := (($assign_spat($sound_name))[2])
  $spat_type := ("to_" + (($assign_spat($sound_name))[1]))
  $dd := ($d*$factDist)
  @command($spat_type) source $source ade $a $dd $e
}

; -- ::spat_traj_ade($sound_name, [[[$a1, $d1, $e1], [$a2, $d2, $e2], $interp1, $dur1], [[[$a3, $d3, $e3], [$a4, $d4, $e4], $interp2, $dur2]])
@proc_def ::spat_traj_ade($sound_name, $tab)
{
  @local $tab_args, $dur_segment

  $tab_args := $tab[0]
  $dur_segment := $tab_args[3]
  if (@size($tab) == 1) { ::spat_traj_ade_($sound_name, $tab_args) }
  else {
    ::spat_traj_ade_($sound_name, $tab_args)
    $dur_segment := spat_traj_ade($sound_name, @cdr($tab))
  }
}

; -- ::spat_traj_ade_($sound_name, [[[$a1, $d1, $e1], [$a2, $d2, $e2], $interp, $dur])
@proc_def ::spat_traj_ade_($sound_name, $tab_args)
{
  @local $source, $spat_type, $aa, $dd, $ee, $a1, $a2, $d1, $d2, $e1, $e2, $interp, $dur_segment

  ; arguments du proc
  $a1 := $tab_args[0,0]
  $d1 := $tab_args[0,1]
  $e1 := $tab_args[0,2]
  $a2 := $tab_args[1,0]
  $d2 := $tab_args[1,1]
  $e2 := $tab_args[1,2]
  $interp := $tab_args[2]
  $dur_segment := $tab_args[3]

  $source := (($assign_spat($sound_name))[2])
  $spat_type := ("to_" + (($assign_spat($sound_name))[1]))
  curve traj @grain := $grain_spat s, @Action := @command($spat_type) source $source ade $aa ($dd*$factDist) $ee
  {
    $aa, $dd, $ee
    {
      $dur_segment { $a1 $d1 $e1 } @type $interp
      $dur_segment { $a2 $d2 $e2 }
    }
  }
}

```



### ANNEXE 3

```

; -- ::spat_traj_xye_($sound_name, [[$x1,$y1,$e1], [$x2,$y2,$e2], $interp, $dur])
@proc_def ::spat_traj_xye($sound_name, $tab)
{
    @local $tab_args, $dur_segment

    $tab_args := $tab[0]
    $dur_segment := $tab_args[3]
    if (@size($tab) == 1) { ::spat_traj_xye_($sound_name, $tab_args) }
    else {
        ::spat_traj_xye_($sound_name, $tab_args)
        $dur_segment := spat_traj_xye($sound_name, @cdr($tab))
    }
}

@proc_def ::spat_traj_xye_($sound_name, $tab_args)
{
    @local $source, $spat_type, $xx, $yy, $ee, $x1, $x2, $y1, $y2, $e1, $e2, $interp, $dur_segment

    ; arguments du proc
    $x1 := $tab_args[0,0]
    $y1 := $tab_args[0,1]
    $e1 := $tab_args[0,2]
    $x2 := $tab_args[1,0]
    $y2 := $tab_args[1,1]
    $e2 := $tab_args[1,2]
    $interp := $tab_args[2]
    $dur_segment := $tab_args[3]

    $source := (($assign_spat($sound_name))[2])
    $spat_type := ("to_" + (($assign_spat($sound_name))[1]))
    curve traj @grain := $grain_spat s, @Action := @command($spat_type) source $source xyz $xx $yy $ee
        { $xx, $yy, $ee
            {
                { $x1 $y1 $e1 } @type $interp
                $dur_segment { $x2 $y2 $e2 }
            }
        }
}

; -- ::spat_presence($sound_name, $presence)
@proc_def ::spat_presence($sound_name, $presence)
{
    @local $source, $spat_type, $a, $d, $e

    $source := (($assign_spat($sound_name))[2])
    $spat_type := ("to_" + (($assign_spat($sound_name))[1]))

    @command($spat_type) source $source pres $presence
}

; pour tester ...
; $spat_son1 := obj::spat("son1", 10, "csound1", "hoa", ::spat_traj_ade, [[0,1,0,320,0.5,0,"linear",10]])
; $spat_son2 := obj::spat("son2", 10, "csound2", "hoa", ::spat_traj_fun, [:lissaj5,1,20,0,0])
; @current_assign_tab("hoa")
;
; 5 $spat_son3 := obj::spat("son3", 7, "csound1", "hoa", ::spat_traj_xye, [[-1,1,0,1,-1,0,"linear",7]])
; $spat_son4 := obj::spat("son4", 10, "csound2", "hoa", ::spat_traj_fun, [:lissaj4,1,20,0,0])
; @current_assign_tab("hoa")

```

### ANNEXE 3

```

; -----
; --- MACROS DE SPATIALISATION --- voir schémas -----
;
$pi := 3.1415926535897
; -----> SPAT COURBE1 % RT_TEMPO
; la vitesse détermine le temps de la courbe complète

; une sorte de papillon favorisant avant-arrière (un coup L un coup R)
@proc_def ::lissaj1($SPAT,$sourceNum,$distanceL1,$vitesse1,$offset,$elevation)
{
  @local $liss1x
  $liss1x := 0.
  loop LISSAJOU1 $vitesse1
  {
    ;abort liss1
    curve liss1 @grain := $grain_spat s, @Action := @command($SPAT) source $sourceNum xyz (@cos((
      $liss1x+$offset)*2.)*($distanceL1*$factDist)) (@sin($liss1x*5.)*($distanceL1*$factDist))
      $elevation
      { $liss1x
        {
          $vitesse1 { $liss1x }
        }
      }
    }
  }

; la version plus circulaire avec passage à côté du milieu
@proc_def ::lissaj2($SPAT,$sourceNum,$distanceL2,$vitesse2,$offset,$elevation)
{
  @local $liss2x
  $liss2x := 0.
  loop LISSAJOU2 $vitesse2
  {
    ;abort liss2
    curve liss2 @grain := $grain_spat s, @Action := @command($SPAT) source $sourceNum xyz (@cos((
      $liss2x+$offset)*3.)*($distanceL2*$factDist)) (@sin($liss2x*5.)*($distanceL2*$factDist))
      $elevation
      { $liss2x
        {
          $vitesse2 { $liss2x }
        }
      }
    }
  }

@proc_def ::lissaj2inv($SPAT,$sourceNum,$distanceL2,$vitesse2inv,$offset,$elevation)
{
  @local $liss2invx
  $liss2invx := 6.2831853071
  loop LISSAJOU2 $vitesse2inv
  {
    ;abort liss2inv
    curve liss2inv @grain := $grain_spat s, @Action := @command($SPAT) source $sourceNum xyz (@cos((
      $liss2invx+$offset)*3.)*($distanceL2*$factDist)) (@sin($liss2invx*5.)*($distanceL2*$factDist))
      $elevation
      { $liss2invx
        {
          $vitesse2inv { ($liss2invx-6.2831853071) }
        }
      }
    }
  }

; celui-là fait une diagonale en passant au milieu, donc à utiliser avec aperture = 180
@proc_def ::lissaj3($SPAT,$sourceNum,$distanceL3,$vitesse3,$offset,$elevation)
{
  @local $liss3x
  $liss3x := 0.
  loop LISSAJOU3 $vitesse3
  {
    ;abort liss3
    curve liss3 @grain := $grain_spat s, @Action := @command($SPAT) source $sourceNum xyz (@cos((
      $liss3x+$offset)*3.)*($distanceL3*$factDist)) (@sin($liss3x*4.)*($distanceL3*$factDist))
      $elevation
      { $liss3x
        {
          $vitesse3 { $liss3x }
        }
      }
    }
  }
}

```

# A tutorial for composers

JBL0\_examples
Language Ex Example 1 Step 1 Step 2 Step 3 Example 2 Step 1 Step 2

## Antescofo language examples

Before you begin, make sure your DSP is ON and Overdrive is ON in Audio Status

- In these examples, we will explore different compositional manners of using Antescofo language elements such as: loops, groups, curves processes, and conditional actions
- This is to show how such simple constructions can allow interesting and sometimes complex compositional processes such as tempo modulations, high-level control of sound synthesis, and more generally a compositional environment for Mixed Music pieces and also pure electronic music.

We will proceed with two examples which employ very simple synthesis patches in Max, made purely for demonstrating language possibilities offered by Antescofo.

- The first example uses Antescofo "loops" and "curves" and demonstrates a simple tempo modulation and a simple spatial control which can sometimes be heavy to program in Max. Advanced users will also discover particular uses of Antescofo tables for creating rhythmic articulations.

Preview Example 1 here ==>>> start play stop

- The second example uses Antescofo "processes" and "curves" to drive a sound synthesis soundscape generated stochastically using predefined envelopes.

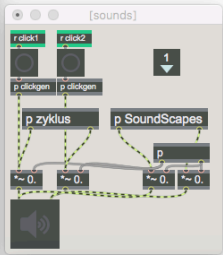
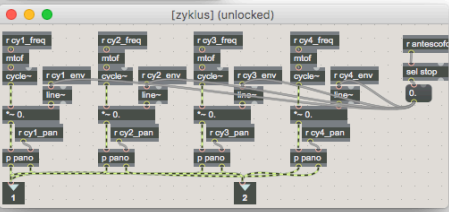
Preview Example 2 here ==>>> start play stop

**Use the tabs above to explore details of each example!**  
Prepared by Julia Blondeau: <http://www.juliablondeau.fr/>

r antescofo-mess

antESCOfo-

p sounds

JBL0\_examples
Language Ex Example 1 Step 1 Step 2 Step 3 Example 2 Step 1 Step 2

## Example 1

Example 1 : loops, curves, tempo modulation...

start
play
stop

The first example contains 2 "layers" which we will study separately in Steps 1 and 2 above:

- the first is composed with 2 pulsations (a click and plus an additive synthesis attack) with a tempo modulation and some conditional actions.
- the second layer is a synthesized "SoundScape" where the envelope are generated with multiple curves.

You can scroll the bar to hear just the first layer, the second or the twice.

Layer 1

Layer 2

Time to get into step-by-step details...  
**Go to Step One !**


Click to see the score in AscoGraph

JBL0\_examples
Language Ex Example 1 Step 1 Step 2 Step 3 Example 2 Step 1 Step 2

### Example 1, step One :

## Layer 1

a tempo modulation for clicks!



BPM = 100

tempo

BPM = 30

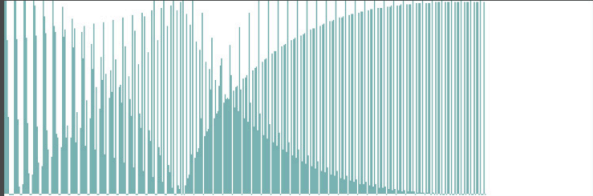
For the first layer, a table of amplitude is employed to represent triple-crochet rhythmic subdivisions. This will allow us to perform interpolations between tables to vary rhythms. Pulsations are controlled by a loop whose termination is defined as a condition on tempo.

The click pulse thus goes through a tempo modulation controlled by a curve through the variable \$t\_loop1. When the tempo curve arrives at value 90, the loop is aborted using Antescofo's "until" construction.

**You have to see the score !**

start
play
stop

=> Antescofo



```

; Open and study this text file...
@INSERT "JBLO_examples_macros.asco.txt"
BPM 60
psound open ; open the "sounds" patcher in help window @local

; tables of rhythmic subdivisions on one beat
$tabR1 := [0.,1.,0.,0.,1.5,0.,1.,1.]
$tabR2 := [1.,0.,0.,1.3,0.,1.,0.,0.]
$tabR3 := [0.,0.,0.,0.,0.4,0.8,1.1,1.3]

; tables of amplitudes (for 5 synthesisengines)
$stabE1 := [ 0.,0.,0.4,0.3]
$stabE2 := [ 0.05,0.4,0.,0.4]
$stabE3 := [ 0.1,0.05,0.3,0.]

$st_loop1 := 60
$sfreqZ := 64

EVENT 30 one ; a dummy score event

;----- LAYER 1 -----
; The first group uses a curve to control the tempo of a loop and an another
; curve to control the passage between tabs who give the amplitude of clics
; and for the synthesis.
GROUP ligne1
{
  curve tempoloop1 @grain := 0.05s, @action := recvr_tempo $st_loop1
  {
    $st_loop1
    {
      5 { 60 } @type "quad"
      15 { 80 }
      7 { 40 } @type "quad_out"
      12 { 90 }
    }
  }
  curve matrixRythm @grain := 0.05s
  {
    $tabR4
    {
      12 { $tabR1 } @type "cubic"
      16 { $tabR2 }
      16 { $tabR3 }
    }
  }
  loop rhythm_process_1 1.0
  @tempo := $st_loop1
  {
    GROUP loop1G1
    {
      ampClic1 ($tabR4[0])
      click1 bang
      1/8 ampClic2 ($tabR4[1])
      click2 bang
      1/8 ampClic1 ($tabR4[2])
      click1 bang
      1/8 ampClic2 ($tabR4[3])
      click2 bang
      1/8 ampClic1 ($tabR4[4])
      click1 bang
      1/8 ampClic2 ($tabR4[5])
      click2 bang
      1/8 ampClic1 ($tabR4[6])
      click1 bang
      1/8 ampClic2 ($tabR4[7])
      click2 bang
    }
  } until ($st_loop1 == 90)
}
}

$ccount := 1
curve matrixZyklus @grain := 0.05s ; used for the ADSR macro
{
  $stabE
  {
    { $stabE1 }
    { 12 { $stabE2 }
      16 { $stabE3 }
    }
  }
}

$cycleloop := 1/3
whenever ($st_loop1 > 60 )
{ let $cycleloop := 1/3 }
whenever ($st_loop1 < 60 )
{ let $cycleloop := 1/5 }

loop Zyklus $cycleloop
@tempo := $st_loop1
{
  @command(("cy"+$ccount+"_freq")) ((random()) + $sfreqZ)
  @ADSR($ccount,10,30,90,60)
  @command(("cy"+$ccount+"_pan")) (@random()-0.5)
  if ($ccount >= 3)
  { let $ccount := 1 }
  else { let $ccount := $ccount + 1 }
} until ($st_loop1 == 90)

$span4 := -1
8 loop Zyklus2 1
{
  cy4_freq 88
  @ADSR(4,10,10,40,200)
} during [20]

loop pan4 6
{
  curve slider @Grain := 0.05s, @Action := cy4_pan $span4
  {
    $span4
    {
      { -1. }
      3 { 1. }
      3 { -1. }
    }
  }
} during [23]
}

```

```

;----- LAYER 2 -----
; The second group, in parallel with the previous one, use a curve to control all
; layers of the synthesis and to control the panning of a kind of "sonar" !
GROUP ligne2
{
  curve slider @Grain := 0.05s, @Action := {
    L2-1_env $11 30
    L2-2_env $12 30
    L2-3_env $13 30
    L2-4_env $14 30
    L2-5_env $15 30
  }
  {
    $11, $12, $13, $14, $15
    {
      { 0. 0. 0. 0. 0. } @type "quad"
      6 { 0.02 0.02 0.01 0.03 0.02 }
      2 { 0.09 0.002 0.1 0.003 0.002 }
      2 { 0.002 0.06 0.001 0.09 0.11 }
      2 { 0.05 0.09 0.12 0.05 0.06 }
      2 { 0.11 0.002 0.001 0.09 0.002 }
      2 { 0.02 0.07 0.11 0.03 0.08 }
      2 { 0.08 0.002 0.001 0.03 0.002 }
      2 { 0.002 0.06 0.001 0.09 0.11 }
      4 { 0.05 0.09 0.12 0.05 0.06 }
      20 { 0. 0. 0. 0. 0. }
    }
  }
}
}

```

EVENT 5 three ; a dummy score event

Example 1, step Two :  
Layer 1.2  
Alternative rhythms

The second pulsation in this example provides an alternative version of rhythmic clicks seen previously. The same amplitudes stored in tables are this time used in an Antescofo Macro that create an ADSR for a simple additive synthesis engine.

The loop named "Zyklus" sends parameters of the additive synthesis to Max patches. The conditional action "if" allows changing of the parameter destination. The internal function "@command" allows dynamic construction of receive symbols in Antescofo actions.

In this example, we see the possibility of refining each cycle of a loop dynamically. This is achieved by conditional actions which are themselves dynamic. When the tempo is greater than 60, the cycle is equal to a triplet. When the tempo is less than 60, it becomes equivalent to a quintolet. To achieve this, we use the "whenever" command. In contrast to "if", "whenever" remains always active and parallel to other actions in Antescofo which is what we need here.

Example 1, step Three :  
Layer 2  
A multi-curves landscape and a sonar...

In this step, we will add a third layer which is simply continuous amplitude controls for the SoundScape patcher of this example. We employ a multi-dimensional curve to control amplitudes!

envelopes of 5 layers of synthesis

JBLO\_examples

Language Ex Example 1 Step 1 Step 2 Step 3 **Example 2** Step 1 Step 2

# Example 2

In processus we trust


Our second example demonstrate the power of Antescofo "Processes" that allows dynamic instantiation of 5 synthesis voices using continuous controls. Each process is launched stochastically defining the envelope and the duration of each synthesis engine.

Here we create the same synthesis soundscape as before. But this version allows you to create as many instants using the same definition, with each with different values. Therefore Antescofo processes allow modular authorship of electronic processes.

We will see a modular programming where each envelope is generated by a process. A 'general' process will be used to read envelopes and to choose randomly the duration of each voice envelope.

start play stop  
=> Antescofo

Open AscoGraph


Go to Step One ! 

JBLO\_examples

Language Ex Example 1 Step 1 Step 2 Step 3 Example 2 **Step 1** Step 2

## Example 2 :

# Step One

Envelop Library... 

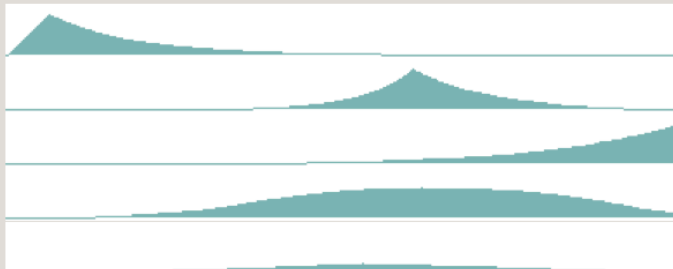
We start by creating 5 processes each containing a particular envelope type. As arguments, they accept the destination number of the synthesis engine and also the duration for the envelope.

We launch the 5 processes in parallel and obtain 5 synthesis voices that are superposed. Each envelope here is approximately 18 seconds.

start play stop  
=> Antescofo

Open AscoGraph

**envelopes of 5 layers of synthesis**



# ANNEXE 3

```
BPM 60
| psound open ; open the "sounds" patcher in help window @local
```

```
; variables initialization
```

```
$l1 := 0.
$l2 := 0.
$l3 := 0.
$l4 := 0.
$l5 := 0.
sl := [$l1,$l2,$l3,$l4,$l5]
```

```
l2-1_env 0. 30
l2-2_env 0. 30
l2-3_env 0. 30
l2-4_env 0. 30
l2-5_env 0. 30
```

```
@proc_def ::env_1($NumLine,$dur)
{
  @local $lenv1
  $lenv1 := $l[($NumLine-1)]
  curve envLine1 @grain := 0.05s, @action := { $l[($NumLine-1)] := $lenv1
    @command("l2-"+$NumLine+"_env") $lenv1 }
  { $lenv1
    { (15*($dur/16)) { $lenv1 } @type "exp"
      ($dur/16) { 0. }
    }
  }
}
```

```
@proc_def ::env_2($NumLine,$dur)
{
  @local $lenv2
  $lenv2 := $l[($NumLine-1)]
  curve envLine2 @grain := 0.05s, @action := { $l[($NumLine-1)] := $lenv2
    @command("l2-"+$NumLine+"_env") $lenv2 }
  { $lenv2
    { (4*($dur/8)) { $lenv2 } @type "exp"
      (4*($dur/8)) { 0. } @type "exp_out"
    }
  }
}
```

```
@proc_def ::env_3($NumLine,$dur)
{
  @local $lenv3
  $lenv3 := $l[($NumLine-1)]
  curve envLine3 @grain := 0.05s, @action := { $l[($NumLine-1)] := $lenv3
    @command("l2-"+$NumLine+"_env") $lenv3 }
  { $lenv3
    { ($dur/16) { $lenv3 }
      (15*($dur/16)) { 0. } @type "quart_out"
    }
  }
}
```

```
@proc_def ::env_4($NumLine,$dur)
{
  @local $lenv4
  $lenv4 := $l[($NumLine-1)]
  curve envLine4 @grain := 0.05s, @action := { $l[($NumLine-1)] := $lenv4
    @command("l2-"+$NumLine+"_env") $lenv4 }
  { $lenv4
    { (4*($dur/8)) { $lenv4 } @type "quad_in_out"
      (4*($dur/8)) { 0. } @type "quad_in_out"
    }
  }
}
```

```
@proc_def ::env_5($NumLine,$dur)
{
  @local $lenv5
  $lenv5 := $l[($NumLine-1)]
  curve envLine5 @grain := 0.05s, @action := { $l[($NumLine-1)] := $lenv5
    @command("l2-"+$NumLine+"_env") $lenv5 }
  { $lenv5
    { (4*($dur/8)) { $lenv5 }
      (4*($dur/8)) { 0. }
    }
  }
}
```

```
$stabProc := [::env_1,::env_2,::env_3,::env_4,::env_5]
```

```
@proc_def ::launcher($NumLine)
{
  @local $dur, $NumProc
  $NumProc := @rand_int(@size($stabProc))
  $dur := ((@rand_int(11))+1)
  :: ($stabProc[$NumProc])($NumLine,$dur)
  ($dur+0.001) ::launcher($NumLine)
}
```

```
EVENT 50
curve slider @Grain := 0.05s, @Action := { l2-1_env $l1 30
  l2-2_env $l2 30
  l2-3_env $l3 30
  l2-4_env $l4 30
  l2-5_env $l5 30
}
{
  $l1, $l2, $l3, $l4, $l5
  {
    { 0. 0. 0. 0. 0. } @type "quad"
    2 { 0.02 0.02 0.01 0.03 0.02 }
  }
}

; The 5 voices of synthesis
2 ::launcher(1)
::launcher(2)
::launcher(3)
::launcher(4)
::launcher(5)

EVENT 10
abort ::launcher
curve slider @Grain := 0.05s, @Action := { l2-1_env $l1 30
  l2-2_env $l2 30
  l2-3_env $l3 30
  l2-4_env $l4 30
  l2-5_env $l5 30
}
{ $l1, $l2, $l3, $l4, $l5
  {
    { $l1 $l2 $l3 $l4 $l5 } @type "quad_out"
    2 { 0. 0. 0. 0. 0. }
  }
}
```

JBLO examples

Langage Ex    Exemple 1    Step 1    Step 2    Step 3    Exemple 2    Step 1    **Step 2**

Example 2 :  
*Step Two*  
The infinite machine...

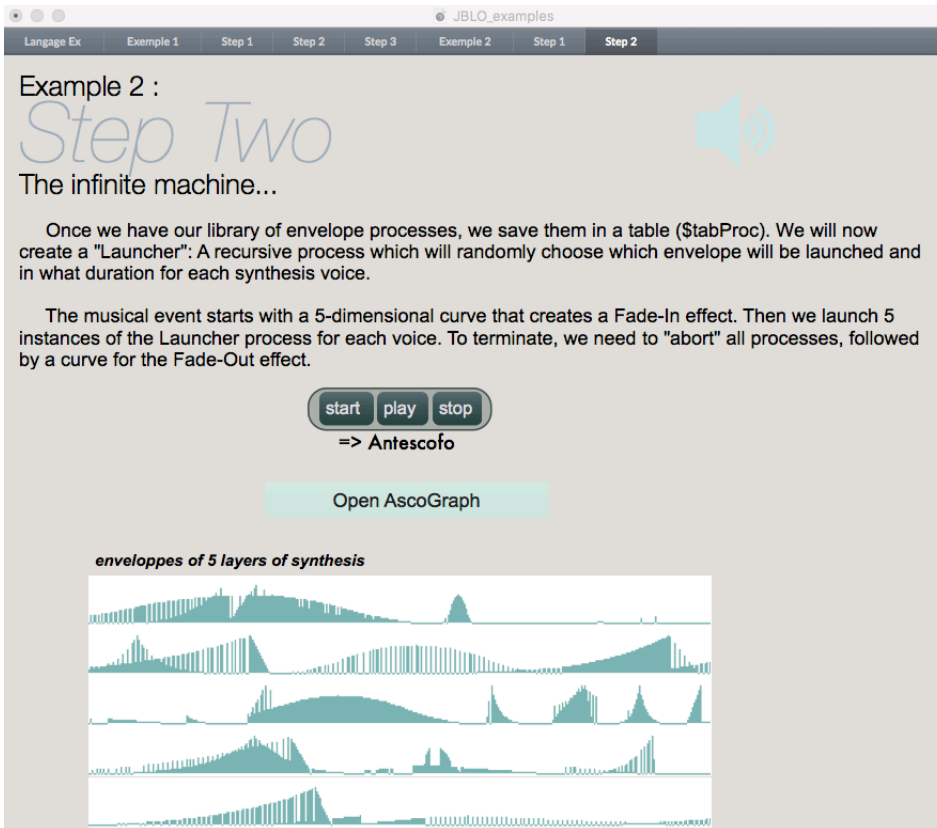
Once we have our library of envelope processes, we save them in a table (\$tabProc). We will now create a "Launcher": A recursive process which will randomly choose which envelope will be launched and in what duration for each synthesis voice.

The musical event starts with a 5-dimensional curve that creates a Fade-In effect. Then we launch 5 instances of the Launcher process for each voice. To terminate, we need to "abort" all processes, followed by a curve for the Fade-Out effect.

start   play   stop  
=> Antescofo

Open AscoGraph

*envelopes of 5 layers of synthesis*



## BIBLIOGRAPHIE

Georgio **AGAMBEN**, *Signatura rerum*. Editions Vrin.

Andrea **AGOSTINI**, Daniele **GHISI** et Eric **MAESTRI**, *Recreating Gérard Grisey's Vortex Temporum with cage*, dans Proceedings of the International Computer Music Conference (ICMC), Utrecht. 2016

Philippe **ALBERA**, *Le son et le sens, Essais sur la musique de notre temps*. Editions Contrechamps, 2007.

Rudolf **ARNHEIM**, *Visual Thinking*. University of California Press, 1969. Trad française : « La pensée visuelle », Champs-Flammarion, 1976.

Ron **ATKIN**, *Combinatorial Connectivities in Social Systems: an application of simplicial complex structures to the study of large organizations*. (Vol. 34). Birkhauser, 1977.

Ron **ATKIN**, *Multidimensional Man: Can Man Live in 3-dimensional Space ?*. Penguin Press, 1981.

Gérard **BAILLET**, *Gérard Grisey, fondements d'une écriture*. Editions l'Harmattan, 2000.

Walter **BENJAMIN**, *Images de pensée*, Paris : Christian Bourgeois Éditeur, 1998.

Claude **BERROU** & Vincent **GRIPON**. *Petite mathématique du cerveau*. Éditions Odile Jacob, 2012.

José-Luis **BESSADA**, *Musique, calcul (cognitif) secret ? Alberto Posadas ou l'échappée hors du formalisme*. Sur : <http://brahms.ircam.fr/documents/document/21540/>

Louis **BIGO**, Moreno. **ANDREATTA**, J.-L. **GIAVITTO**, Olivier **MICHEL**, and Antoine **SPICHER**. *Computation and Visualization of Musical Structures in Chord-based Simplicial Complexes*, Mathematics and Computation in Music (MCM 2013, Montreal), LNCS 7937, p38-51, 2013.

Pierre **BOULEZ**, *Penser la musique aujourd'hui*. Gallimard, collection Tel n°124, 1987.

Pierre **BOULEZ**, *Leçons de musique. Points de repère III*. Éditions Christian Bourgeois, 2005.

Jean **BRESSON**, Carlos **AGON**, and Gérard **ASSAYAG**. *Visual lisp/clos programming in openmusic*. Higher-Order and Symbolic Computation, 22(1) :81–111, 2009.

Jean-Pierre **BRIOT**, Gaétant **HADJERES**, François **PACHET**, *Deep learning techniques for music generation — A survey*. Arxiv:1709.01 620, septembre 2017.

Jean-Pierre **CHANGEUX**, *L'Homme neuronal*. Fayard, 1983.

Arshia **CONT**, José **ECHEVESTE**, Florent **JACQUEMARD**, Jean-Louis **GIAVITTO**. *Correct automatic accompaniment despite machine listening or human errors in Antescofo*. ICMC 2012 (International Computer Music Conference), 9-15 September, Ljubljana, Slovenia.

François **DECARSIN**. *La musique, architecture du temps*. L'Harmattan, 2001.

François **DECARSIN**, *Karlheinz Stockhausen*, dans *Théories de la composition musicale au XX<sup>ème</sup> siècle*,



BIBLIOGRAPHIE

vol.2. Collection Symétrie Recherche, série 20-21, 2013.

Gilles **DELEUZE**, *Logique du sens*. Paris, Éditions de Minuit, 1979.

Gilles **DELEUZE**, *Différence et répétition*. Presses Universitaires de France - PUF, 2011.

Gilles **DELEUZE** et Félix Guattari, *Mille Plateaux, Capitalisme et schizophrénie 2*. Les Éditions de Minuit, 1980.

Georges **DIDI-HUBERMAN**. *Atlas, ou le gai savoir inquiet*. T. L'oeil de l'histoire, 3. Les Éditions de Minuit, 2011.

Georges **DIDI-HUBERMAN**. *L'image survivante, histoire de l'art et temps des fantômes selon Aby Warburg*. Les Éditions de Minuit, 2002.

Georges **DIDI-HUBERMAN**. *Quand les images prennent position*. T. L'oeil de l'histoire, 1. Les Éditions de Minuit, 2009.

Hugues **DUFOURT**, *L'artifice d'écriture dans la musique occidentale*. Critique, 408, 1981.

Hugues **DUFOURT**, *L'ordre et le sensible*. Dans : *Quoi ? Quand ? Comment ? : La recherche musicale (1985)*.

Hugues **DUFOURT**, *Musique, pouvoir, écriture*. C. Bourgeois, Paris, 1991.

Jacques **DUTHEN** et Marco **STROPPA**, *Une représentation de structures temporelles par synchronisation de pivots*. In Proceedings of Colloque « Musique et Assistance Informatique », pages 471–479, 1990.

José **ECHEVESTE**, *Un langage de programmation pour composer l'interaction musicale*. Thèse de doctorat. 2015.

José **ECHEVESTE**, Arshia **CONT**, Jean-Louis **GIAVITTO** and Florent **JAQUEMARD**. *Operational semantics of a domain specific language for real time musician–computer interaction*. J. of Discrete Event Dynamic Systems, vol. 23, n°4, pp 343-383, 2013.

Laurent **FENEYROU**, *Pierre Boulez*, dans *Théories de la composition musicale au XX<sup>ème</sup> siècle*, vol.2. Collection Symétrie Recherche, série 20-21, 2013.

Michel **FOUCAULT**, *Des espaces autres*. Dans : *Dits et écrits*. Sous la dir. de Gallimard NRF. 1984.

Michel **FOUCAULT**, *Les mots et les choses*, vol. 166. Tel, Gallimard, 1966.

Maurice **FRÉCHET** et Ky **FAN**, *Introduction à la topologie combinatoire*. Paris, Éditions Jacques Gabay, 1946.

Jean-Louis **GIAVITTO**. *Du temps écrit au temps produit en informatique musicale*. Dans : Ircam, «Produire le temps» (2012).

Jean-Louis **GIAVITTO**. *Topological collections, transformations and their application to the modeling and the simulation of dynamical systems*. In *Rewriting Technics and Applications (RTA'03)*, volume LNCS 2706 of LNCS, pages 208 – 233, Valencia, June 2003. Springer.

Jean-Louis **GIAVITTO** and Antoine **SPICHER**. *Topological rewriting and the geometrization of programming*. Physica D, 237(9) :1302–1314, jully 2008.

Jean-Louis **GIAVITTO** and Antoine **SPICHER**. *Morphogenèse*, chapter Morphogenèse informatique, pages 178–198. Belin, 2006.

Jean-Louis **GIAVITTO** and Erica **VALENCIA**. *Diagrammatic Representation and Reasoning*, chapter A Topological Framework for Modeling Diagrammatic Reasoning Tasks. Springer-Verlag, 2001.

## BIBLIOGRAPHIE

- Gérard **GRISEY**, *Zur Entstehung des Klanges...*, *Darmstädter Beiträge zur neuen Musik*, 17 (1978). Traduction française *Ecrits ou l'Invention de la musique spectrale*, édition établie par Guy Lelong, collection Répercussions, Paris : MF, 2008.
- Maud **HAGELSTEIN**, *Mémoire et Denkraum. Réflexions épistémologiques sur la Kulturwissenschaftliche Bibliothek Warburg*, *Conserveries mémorielles*, #5 | -1.
- Edmund **HUSSERL**, *Leçons pour une phénoménologie de la conscience intime du temps*. Presses Universitaires de France - PUF, 1996.
- Helmut **LACHENMANN**, *Ecrits et entretiens*. Contrechamps, 2009.
- Helmut **LACHENMANN**, *Quatres aspects fondamentaux du matériau musical et de l'écoute*. Dans : *Inharmoniques n8/9*, Paris, IRCAM-Centre Pompidou (1991).
- Carlo **LAURENZI** et Marco **STROPPA**, *The notation of dynamic levels in the performance of electronic music*. Proceedings of Tenor Conference, 2015.
- Mikael **LAURSON**, *PATCHWORK : a Graphic Language in PREFORM*. Ann Arbor, MI : MPublishing, University of Michigan Library, 1989.
- Claude **LEVI-STAUSS**, *La pensée sauvage*, Paris : Plon, 1962
- Fabien **LEVY**, *Complexité grammatologique et complexité aperceptive en musique*. Thèse de doct. EHES, 2004.
- Philippe **MANOURY**, *Les chaînes de Markov... à l'infini*. <http://www.philippemanoury.com/?p=5685>. 2015.
- Philippe **MANOURY**, *Les Grammaires Génératives musicales*. <http://www.philippemanoury.com/?p=5025>. 2012.
- Maurice **MERLEAU-PONTY**, *Phénoménologie de la perception*. Tel, Gallimard, 1976.
- Philippe-Alain **MICHAUD**, *Aby Warburg et l'image en mouvement*, Édition Macula, 1998.
- Geoffroy **PEETERS**, B. **GIORDANO**, P. **SUSINI**, N. **MISDARIIS**, St. **MCADAMS**. *The Timbre Toolbox: Audio descriptors of musical signals*. JASA, 130 (5) 2011
- Henri Poincaré, *Dernières pensées* (Bibliothèque Philos. scient.). Paris, Ernest Flammarion, 1913.
- Angelo **ORCALLI**, *La pensée spectrale*, dans *Théories de la composition musicale au XX<sup>ème</sup> siècle*, vol.2. Collection Symétrie Recherche, série 20-21, 2013.
- Michel **RIGONI**, *Un vaisseau lancé vers le ciel*. Millénaire III éditions, 1998.
- Francesca **ROSSI**, Peter van **BEEK** et Toby **WALSH**, *Handbook of Constraint Programming*. Elsevier.
- Ferdinand de **SAUSSURE**, *Cours de linguistique générale*. Éditions Bally et Sechehaye, (1971). 1916.
- Jean-Paul **SARTRE**, *L'Être et le Néant : Essai d'ontologie phénoménologique*. Paris, Gallimard, coll. « Tel », 1976.

## BIBLIOGRAPHIE

Fritz **SAXL**, *The History of Warburg's Library*, 1944, dans *Gombrich, Aby Warburg. An Intellectual Biography*, Londres : The Warburg Institute (2<sup>ème</sup> éd. 1986)

Diemo Schwarz. *Interacting with a Corpus of Sounds*. eContact! online journal of the Canadian Electroacoustic Community, vol. 2, n° 16.

Karlheinz **STOCKHAUSEN**, *Texte zur elektronischen und instrumentalen Musik. Texte zur Musik*, vol. I., édition Dieter Schnebel. 1963.

Karlheinz **STOCKHAUSEN**, *Comment passe le temps ?* Dans : Contrechamps n° 9 (1988).

Marco **STROPPIA**, *Organismes d'information musicale : une approche de la composition*. La musique et les Sciences Cognitives, Contemporary Music Review, Pierre Mardaga, Bruxelles, 1989.

Marco **STROPPIA**, *Connaître des beautés insoupçonnées*. Entretiens du festival Présences. 2016.

Andrei **TARKOVSKI**, *Le temps scellé. § La responsabilité de l'artiste*. Éditions de l'étoile, Cahiers du cinéma, 1989.

Heinrich **TAUBE**, *Common music : A music composition language in common lisp and clos*. Computer Music Journal, pages 21–32, 1991.

Erika **VALENCIA**, **GIAVITTO**, J. L., & **SANSONNET**, J. P. (1998, August). *Algebraic Topology for Knowledge Representation in Analogy Solving*. In ECAI.

Erika **VALENCIA**, **GIAVITTO**, J. L., & **SANSONNET**, J. P. (1998). *Esqimo: Modelling analogy with topology*. In *Second European Conference on Cognitive Modelling (ECCM2)*, p212-213.

Hugues **VINET**. *Produire le temps*, édité par H. Vinet. Hermann avril 2014. 238p. Contributeurs : Yves André, Gérard Berry, Antoine Bonnet, Nicolas Donin, Laurent Feneyrou, Patrick Flandrin, Jean-Louis Giavitto, Philippe Manoury, François Nicolas, Thierry Paul, François Regnault, Pierre-André Valade, Hugues Vinet.

Ludwig **WITTGENSTEIN**, *Tractatus logico-philosophicus*, Paris : Gallimard, 1975.

Iannis **XENAKIS**, *Musique Architecture*. Tournai : Casterman. 1971.

# PORTFOLIO



## PORTFOLIO

**NAMENLOSEN** (2017), pour 4 solistes (fl, hb,tp,vla), grand ensemble et électronique - 20'  
Commande de Françoise et Jean-Philippe Billarant. Coproduction : Ircam/Centre Pompidou,  
Ensemble Intercontemporain et Philharmonie de Paris.  
Création le 9 juin 2017 par l'Ensemble Intercontemporain, direction : Matthias Pintscher  
Dans le cadre du festival Manifeste 2017.

**SORTIR DU NOIR** (2016), pour violoncelle et électronique - 16'30  
Commande du GMEM - CNCM Marseille (festival «Les Musiques»)  
Création par Séverine Ballon

**THÉÂTRE D'OMBRES** (2015), pour 6 percussions - 15'  
Commande des Percussions de Strasbourg

**TESLA OU L'EFFET D'ÉTRANGETÉ** (2014), pour alto solo et électronique - 19'  
Commande du festival Musica. Coproduction : Ircam-Centre Pompidou  
Création par Christophe Desjardins

**NACHLEBEN** (2014), pour soprano, baryton, percussions et électronique - 8'30  
Créé dans le cadre de l'académie Manifeste par Donatienne Michel-Dansac, Lionel Peintre et  
Richard Dubelski.

**ÉTAT D'EXCEPTION** (2014), pour piano et électronique - 8'  
Créé dans le cadre du Summer Composition Institute de Harvard.

**TESLA** (2014), pour alto solo, ensemble(12) et électronique - 24'  
Créé par Christophe Desjardins et l'ensemble XX-21, direction : Fabrice Pierre.

*Projets à venir...*

**DUO POUR VIOLON ET VIOLONCELLE (2018)**

Commande de Radio France - émission « Création mondiale »  
Ensemble Court-Circuit

**SOPRANO, TROMPETTE, CLARINETTE BASSE, PERCUSSIONS ET ÉLECTRONIQUE (2019)**

Commande du festival de Witten  
Sarah Maria Sun et les solistes de l'ensemble MusikFabrik

**QUATUOR À CORDES ET ÉLECTRONIQUE (2020)**

Quatuor Diotima

## *Conférences*

- **2017 CONFÉRENCE PHILHARMONIE DE PARIS**, Entretien autour de la création de *Namenlosen* à la Philharmonie de Paris, avec Arnaud Merlin (France Musique)
- **2016 INTERNATIONAL COMPUTER MUSIC CONFÉRENCE (ICMC 2016)**, Participation et présentation de la pièce « Tesla » à ICMC 2016, Utrecht
- **2016 COLLEGIUM MUSICÆ** - conférence donnée sur « Espaces compositionnels et temps multiples »
- **2014 FESTIVAL MUSICA**, Strasbourg (02/10/14) : Colloque sur le temps-réel organisé par Philippe Manoury. « Des constructions temporelles à l'interprétation électronique : ce qu'apporte aujourd'hui le temps-réel à la composition ».
- **IRCAM**, Paris : «Séminaire Recherche & Création», (19/05/14) : Présentation de la pièce pour alto, ensemble et électronique en temps-réel « Tesla ou l'effet d'étrangeté ».
- **ECOLE NORMALE SUPÉRIEURE / CNSMD DE LYON**, (24/04/14) : « Du temps écrit au temps produit »
- **UNIVERSITÉ D'AVIGNON** (28/01/14) : « Atlas, intervalle, montage : esquisse d'un espace compositionnel » - Séminaire
- **CARTOMUSE**, organisé par le réseau MaMuX et le Réseau National des Systèmes Complexes (RNSC).
- **IRCAM**, Paris : «Séminaire MaMuX», (27/01/13) : « Notation et espaces compositionnels : processus et enjeux musicaux » - Séminaire sur les nouveaux espaces de la notation.



