



HAL
open science

Analyse des musiques d'informatique, vers une intégration de l'artefact : propositions théoriques et application sur Jupiter (1987) de Philippe Manoury

Maxence Larrieu

► To cite this version:

Maxence Larrieu. Analyse des musiques d'informatique, vers une intégration de l'artefact : propositions théoriques et application sur Jupiter (1987) de Philippe Manoury. Art et histoire de l'art. Université Paris-Est, 2018. Français. NNT : 2018PESC0007 . tel-01757277

HAL Id: tel-01757277

<https://theses.hal.science/tel-01757277v1>

Submitted on 3 Apr 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.



École doctorale Cultures et Sociétés

Thèse de doctorat

Discipline : Musique

Maxence LARRIEU

ANALYSE DES MUSIQUES D'INFORMATIQUE,
VERS UNE INTÉGRATION DE L'ARTEFACT

Propositions théoriques et application sur Jupiter (1987) de Philippe Manoury

Thèse dirigée par

Kevin DAHAN et Martin LALIBERTÉ

Soutenue le

12 01 2018

Jury

Mme. Myriam DESAINTE-CATHERINE

M. Alain BONARDI

M. Kevin DAHAN

M. Martin LALIBERTÉ

Table des matières

Résumé	15
Abstract	18
Remerciements	21
Introduction	25

Première partie : Approche

Chapitre I Points de singularités des artefacts	33
Résumé.....	34
1 Points de singularités.....	35
1.1 Exemple d'artefact pédagogique.....	35
1.1.1 Jean-Claude Risset et Music V.....	35
1.1.2 Charles Dodge et la théorie.....	36
1.1.3 Miller Puckette et Pure Data.....	37
1.2 Singularité : des apparences.....	39
1.3 Singularité : des connaissances.....	39
1.3.1 Conventions formelles.....	40
1.3.2 Synthèses et traitements de signaux.....	40
1.3.3 Recettes de fabrication.....	41
1.4 Singularité : du design.....	42
2 Points de singularités <i>bis</i>	43
2.1 <i>Inharmonique</i> , section IV.....	43
2.1.1 L'instrument.....	43
2.1.2 La « partition ».....	45
2.2 Singularité : sémantique.....	46
2.3 Singularité : densité.....	47
Chapitre II Système Homme Artefact Son et niveaux d'abstraction	49
Résumé.....	50
1 Système Homme Artefact Son.....	51

1.1	Boîte noire.....	51
1.1.1	Introduction.....	51
1.1.2	Éclairer le fonctionnement par le comportement.....	51
1.1.3	Au-delà de l'ingénierie.....	52
1.2	Le système Homme Artefact Son (HAS).....	52
1.2.1	Système cybernétique.....	52
1.2.2	Du sonore.....	53
1.2.3	Homme, Artefact, Son.....	53
2	Éléments de l'Artefact.....	55
2.1	Signal audionumérique.....	55
2.1.1	Vue technologique.....	55
2.1.2	Un signal audible.....	56
2.2	Module et paramètre.....	56
2.2.1	Le langage historique Music N.....	56
2.2.1.a	Un problème.....	57
2.2.1.b	Une solution.....	57
2.3	Au-delà de la technologie.....	58
2.3.1	L'influence de Music N.....	58
2.3.2	Des Ugens aux modules.....	59
2.3.3	Pour investir l'artefact.....	60
3	Des dimensions.....	61
3.1	Dimensions technologiques.....	61
3.1.1	Donnée ~ audio.....	61
3.1.2	Discret ~ continu.....	62
3.2	Dimensions cognitives.....	63
3.2.1	Symbole, signe et signal.....	63
3.2.2	Symbole et signal.....	65
4	Des niveaux d'abstraction.....	67
4.1	Niveaux d'abstraction et connaissance.....	68
4.2	Niveaux d'abstraction en informatique.....	68
4.2.1	Du transistor au schème.....	69
4.2.2	Music N, schème de représentations.....	70
4.2.3	Une surface dans les niveaux informatiques.....	71
4.3	Niveaux d'abstraction et information.....	72

Deuxième partie : Propositions

Chapitre III	Matériau musical et significations dans le système HAS	76
	Résumé.....	77
1	Matériau musical.....	78
1.1	Introduction.....	78

1.2	Première approche : l'analogie.....	78
1.3	Son physique et son perçu.....	79
1.4	Un son musical.....	80
1.5	Le matériau, dans l'interaction.....	81
1.6	Le matériau, dans le temps.....	82
2	Matériau et système HAS.....	83
2.1	Un corrélat.....	83
2.2	Problème de niveaux.....	84
2.2.1	Paramètres dans la musique sérielle.....	85
2.2.2	Paramètres dans l'artefact.....	86
2.2.2.a	Signification computationnelle.....	87
2.2.2.b	Recouvrement de significations.....	87
2.3	Matériau et jaillissement de significations.....	88
3	Vers une caractérisation dynamique.....	90
3.1	Charles Sanders Peirce et la semiosis.....	90
3.1.1	Introduction.....	90
3.1.2	La semiosis.....	91
3.1.3	Dynamique du signe.....	93
3.2	Des significations dans l'artefact.....	93
3.2.1	Deux processus de signification ?.....	94
3.2.1.a	Homme/Artefact.....	94
3.2.1.b	Homme/Son.....	94
3.2.2	Rencontre de significations.....	95
3.2.3	Variables, paramètres et contrôles.....	96
3.2.3.a	Variables.....	96
3.2.3.b	Paramètres.....	97
3.2.3.c	Contrôles.....	97

Chapitre IV Vers un cadre conceptuel pour la description des artefacts

		99
	Résumé.....	101
	Introduction.....	102
1	Analogie avec un modèle bibliographique.....	103
1.1	Vue sommaire du modèle FRBR.....	103
1.1.1	Contexte d'émergence.....	103
1.1.2	Définition du modèle.....	104
1.1.3	Exemple.....	104
1.2	Analogie.....	106
1.2.1	Un modèle conceptuel.....	106
1.2.2	Schéma.....	107
2	Définition du cadre conceptuel.....	109
2.1	Des Modules.....	109

2.1.1	Définition.....	109
2.1.1.a	Entre paramètre et signal audio.....	109
2.1.1.b	Système HAS.....	109
2.1.2	Types de Modules.....	110
2.1.2.a	Synthèses.....	110
2.1.2.b	Traitements.....	110
2.1.3	Vers des Modules conceptuels.....	110
2.1.4	Abrégé.....	112
2.2	Des Algorithmes.....	112
2.2.1	Définition.....	112
2.2.1.a	Au-dessus des Modules.....	112
2.2.1.b	Système HAS.....	113
2.2.1.c	Des processus temporels.....	113
2.2.2	Abrégé.....	113
2.3	Le routing.....	114
2.3.1	Définition.....	114
2.3.1.a	Routing interne.....	114
2.3.1.b	Routing externe.....	115
2.3.2	Abrégé.....	116
2.4	Processus dynamique, imbrication.....	116
2.5	Représentation.....	117
3	Un concept pour éclairer l'aspect dynamique.....	119
3.1	Introduction.....	119
3.2	Potentiel et instance chez M. Stroppa.....	119
3.3	Potentiel et Actualisation, exposition.....	120
3.3.1	Potentiel sonore.....	120
3.3.1.a	Potentiel et méthode de synthèse.....	121
3.3.2	Actualisation.....	121
3.3.3	Potentiel sonore et actualisation dans le cadre conceptuel.....	122
3.4	Des concepts du langage.....	122
3.4.1	Langue et Parole chez F. de Saussure.....	123
3.4.1.a	Un concept pour investir le langage.....	123
3.4.1.b	Définition.....	124
3.4.1.c	Articulation.....	125
3.4.2	Compétence et Performance chez N. Chomsky.....	125
3.4.2.a	Des grammaires.....	126
3.4.2.b	L'aspect créatif du langage.....	127
3.4.2.c	Définition.....	128
3.4.2.d	Articulation : grammaire, compétence et performance.....	130
3.4.3	Parentés.....	131
3.4.4	Éclairer la dynamique du cadre.....	132
4	Structure de contrôle.....	134
4.1	Regroupement de contrôles.....	134
4.1.1	Utilisations de contrôles.....	135

Chapitre V Analyse Musicale, positionnement	137
Résumé.....	139
1 Analyser la musique électroacoustique.....	140
1.1 Analyser à partir du signal.....	141
1.1.1 Une musique à écouter.....	142
1.1.2 Entre signal physique et écoute subjective.....	143
1.2 Analyser en amont du signal.....	143
1.2.1 Génétique musicale et informatique.....	143
1.2.1.a Génétique et écriture.....	144
1.2.1.b Musique d'informatique et génétique musicale.....	144
1.2.2 La <i>faktura</i>	145
1.2.2.a Présentation.....	145
1.2.2.b <i>Factura</i> et Constructivisme.....	146
1.2.2.c Analyse musicale et <i>faktura</i>	147
2 Spécificité de la musique d'informatique.....	148
2.1 Difficulté de prise en compte.....	148
2.1.1 Du code.....	148
2.1.2 Musique sur bande.....	149
2.1.3 Ontologie.....	150
2.2 Une écriture.....	151
2.2.1 Partition exhaustive chez J.-C. Risset.....	151
2.2.2 Écriture directe chez H. Vaggione.....	152
2.2.3 Écriture.....	154
3 Analyse et musique d'informatique.....	156
3.1 Positionnement.....	156
3.1.1 La <i>faktura</i>	156
3.1.2 La génétique.....	156
3.2 Proposition.....	157
3.2.1 Une analyse entre lecture et écoute.....	157
3.2.2 Une analyse en amont et à partir du signal.....	158

Troisième partie : Application

Chapitre VI Introduction à l'analyse de <i>Jupiter</i>	162
Résumé.....	163
1 Introduction à l'analyse.....	164
1.1 Jupiter.....	164
1.2 Histoire et technologie.....	165
1.2.1 La création avec la 4X.....	165
1.2.2 L'Ircam Signal Processing Workstation (ISPW).....	167
1.2.3 L'ajout des PAF.....	168
1.2.4 Vers Max et Pure Data.....	168

1.3	Matériau pour l'analyse.....	169
1.3.1	Les patches.....	169
1.3.2	Partitions.....	169
1.3.3	Enregistrement.....	169
1.4	Outil pour l'analyse.....	170
2	Vue globale de l'artefact.....	171
2.1	Diffusion électroacoustique.....	171
2.2	Le système de synchronisation.....	171
2.2.1	Des événements.....	173
2.2.2	Copie de la partition instrumentale.....	173
2.2.3	Fonctionnement.....	173
2.2.4	La partition électronique.....	174
2.2.5	Représentation.....	175
2.3	Modules et Algorithmes.....	176
2.3.1	Modules.....	176
2.3.2	Algorithmes.....	176
2.3.2.a	Les interpolations du Sampler.....	176
2.3.2.b	Chapo.....	177
2.3.3	Les Modules dans l'implémentation.....	177
2.3.4	Représentation des Modules et du système de synchronisation	178
2.4	Le routing.....	180
2.4.1	Le routing interne.....	180
2.4.2	Le routing externe.....	181
2.4.3	Routing interne : difficulté de représentation.....	181
2.4.4	Représentation routing externe.....	182
3	Structuration de la pièce.....	185
3.1.1	Sectionnement.....	185
3.1.2	Durée et événements.....	185
 Chapitre VII Modules et Structure de contrôle de l'artefact		187
	Résumé.....	188
1	Modules de synthèse.....	189
1.1	Sampler.....	189
1.1.1	Implémentation audio.....	189
1.1.2	Contrôles.....	190
1.1.3	Utilisation avec <i>samp-note</i>	191
1.1.3.a	Introduction.....	191
1.1.3.b	Aperçu global de l'algorithme.....	192
1.1.3.c	Une carte de fréquences.....	193
1.1.3.d	Des interpolations rythmiques.....	194
1.1.3.e	Conclusion, fichiers d'interpolation et d'ostinato.....	195
1.1.3.f	Représentation des séquences.....	196

1.1.4	Utilisation avec <i>addsamp</i>	197
1.2	Additive.....	198
1.2.1	Description.....	198
1.2.2	Contrôle.....	199
1.2.3	Utilisation.....	199
1.3	Chapo.....	200
1.3.1	Description globale.....	201
1.3.1.a	Algorithme et analyse de la flûte.....	201
1.3.1.b	Oscillateurs.....	201
1.3.1.c	Enveloppe spectrale.....	202
1.3.2	Algorithme.....	204
1.3.3	Interaction flûte et spectre.....	206
1.3.4	Contrôle.....	207
1.3.5	Utilisation.....	207
1.3.5.a	Écriture verticale des fréquences.....	207
1.3.5.b	Écriture horizontale.....	209
1.4	Phase Aligned Formant (Paf).....	211
1.4.1	Des formants.....	211
1.4.2	La méthode de synthèse.....	212
1.4.3	L'implémentation.....	213
1.4.3.a	Signal porteur.....	213
1.4.3.b	Signal modulant.....	214
1.4.3.c	Résultat.....	215
1.4.3.d	Vibrato et Frequency Shifter.....	216
1.4.4	Contrôle.....	216
1.4.5	Connexion au Noise.....	217
1.4.6	Algorithmes.....	217
1.4.7	Utilisations.....	218
2	Modules de traitement.....	219
2.1	Harmonizer.....	219
2.1.1	Description d'une voie.....	219
2.1.2	Contrôle.....	220
2.1.3	Utilisations.....	221
2.2	Frequency Shifter.....	222
2.2.1	Description.....	222
2.2.2	Contrôle.....	223
2.2.3	Utilisations.....	223
2.3	Réverb.....	223
2.3.1	Description.....	223
2.3.1.a	Premier bloc : épaissir le signal.....	224
2.3.1.b	Second bloc : la réverbération.....	225
2.3.2	Contrôle.....	225
2.3.3	Sortie.....	226

2.3.4	Utilisations.....	226
2.4	Noise.....	227
2.4.1	Description globale.....	227
2.4.2	Description détaillée.....	227
2.4.3	Contrôle.....	228
2.4.4	Utilisations.....	228
2.5	Spat.....	229
2.5.1	Simuler un espace bidimensionnel.....	229
2.5.2	Quatre déplacements possibles.....	230
2.5.3	La réverbération.....	231
2.5.4	Contrôle.....	231
3	Structure de contrôle.....	232
 Chapitre VIII Outils informatiques pour l'analyse		 234
	Résumé.....	235
	Préambule.....	236
1	Extraire les données des qlist.....	237
1.1	Un tableau pour présenter les variables.....	237
1.1.1	Aperçu.....	237
1.1.2	Résultat.....	238
1.2	Les partitions des Modules.....	239
1.2.1	Aperçu.....	239
1.2.2	Résultats.....	240
2	Représenter les données du Paf.....	243
2.1	Aperçu.....	243
2.2	Fonctionnement.....	244
2.2.1	Repérer les changements d'état.....	245
2.3	Difficultés.....	245
2.4	Résultat.....	245
3	Le routing.....	246
3.1	Introduction.....	246
3.2	(i) Trouver les (pré-)états des Modules.....	246
3.2.1	Détection manuelle.....	247
3.2.1.a	Sampler.....	247
3.2.1.b	Additive et Chapo.....	247
3.2.2	Détection automatique.....	247
3.2.2.a	Aperçu.....	248
3.2.2.b	Fonctionnement.....	248
3.2.2.c	Difficultés.....	250
3.3	(ii) Extraire le Routing.....	250
3.4	(iii) Dédire les états des Modules et le routing.....	251
3.4.1	Aperçu.....	251

3.4.2	Fonctionnement.....	251
3.4.2.a	Règles.....	252
3.4.2.b	Chaînes de routing et algorithme.....	253
3.4.3	Difficultés.....	253
3.5	Résultat.....	254
3.5.1	Les Modules actifs.....	254
3.5.2	Modules actifs par section.....	255
3.5.3	Les chaînes de routing.....	256
4	Report des résultats.....	258
4.1	Les écritures des Modules.....	258
4.2	Le routing.....	259
 Chapitre IX Analyse de <i>Jupiter</i>		 260
	Résumé.....	261
	Note sur la segmentation.....	262
1	Espace compositionnel.....	263
1.1	Introduction.....	263
1.2	Matériau.....	263
1.2.1	La phrase d'ouverture.....	263
1.2.2	La cellule c.....	264
1.2.3	Note polaire.....	264
1.2.4	Motifs α et β	265
1.2.5	Les triples de l'Harmonizer.....	265
1.2.6	Des agrégats.....	266
1.2.7	Brouiller des sons.....	267
1.3	Archétypes musicaux.....	268
1.3.1	Aérien.....	268
1.3.2	Continuité.....	269
1.3.3	Interaction continue.....	269
1.3.4	Geste proéminent.....	269
1.3.5	Ostinato.....	270
1.4	Deux zones protéiformes.....	270
2	Analyse.....	272
	Section I.....	272
	Section II.....	276
	Section III.....	278
	Section IV.....	281
	Section V.....	282
	Section VI.....	284
	Section VII.....	289
	Section VIII.....	289
	Section IX.....	292

Section X.....	294
Section XI.....	295
Section XII.....	296
Section XIII.....	299
3 Vue globale.....	300
3.1 Sections et archétypes.....	300
3.2 Une structure ?.....	301
3.3 Un maillage de réminiscences.....	301
3.4 De l'expérimentation.....	303
Conclusion	305
Bibliographie	315
Annexes	332
Annexe A L'algorithme d'interpolation, captures d'écran	333
Annexe B Notes pour l'algorithme d'interpolation du Sampler. Extrait de [Manoury, 2017]	339
Annexe C Description de l'algorithme qui extrait les chaînes de routing	340
1 Description textuelle.....	340
2 Description en pseudo-code.....	341
Annexe D Deux curiosités dans le patch	342
Annexe E Indications textuelles dans la partition	343

Résumé

Cette thèse traite du médium des musiques d'informatique dans une perspective d'analyse musicale. Les spécificités de ce médium, numérique, sont utilisées pour questionner la pratique de l'analyse. Après les travaux clés de [Battier, 2003] et [Manning, 2006], et plus fortement en relevant que ces musiques existent avec ces médiums, nous faisons l'hypothèse que ces derniers peuvent être une ressource riche pour l'analyse musicale. La question posée est alors « comment les saisir afin d'éclairer la musique avec laquelle ils sont en lien ? ». Notre réponse se fait en deux temps, d'abord avec des propositions théoriques et ensuite en les mettant en pratique dans le cadre d'une analyse sur la pièce *Jupiter* (1987) de Philippe Manoury, pour flûte, traitements et électronique.

Nos propositions sont précédées d'un positionnement terminologique. Très vite il est apparu que ce que nous étudions n'est pas le *patch*, ou encore le *code*, mais bien plus les idées musicales qui ont régi la réalisation de ces derniers. À ces derniers termes nous avons donc substitué celui d'*artefact*, qui rend compte du pouvoir heuristique de ce médium d'une part, et des idées musicales véhiculées par l'ensemble des lignes de code d'autre part.

La réponse à notre problématique se fait au travers de trois propositions. La première est de concevoir un système incluant l'Homme, l'Artefact et le Son. C'est en effet en considérant l'artefact au sein d'un système, qui appelle d'innombrables interactions, que les idées musicales pourront être atteintes. La deuxième, par voie de conséquence, est de concevoir la composition comme un processus hautement dynamique. Ce processus est éclairé à l'aide de la théorie sémiotique de Charles Sanders Peirce, qui nous permet de mettre en lumière des jaillissements de significations. Corrélativement nous comprenons alors que le travail d'analyse, de saisie de l'artefact, doit se faire au sein d'un processus analogue, où les significations jaillissent au fur et à mesure du travail. La troisième proposition est un cadre conceptuel qui vise à décrire les artefacts. Ce cadre doit permettre dans un premier temps d'orienter la saisie des artefacts et dans un second de les décrire. Il se définit essentiellement avec les termes *variable*, *paramètre*, *contrôle* d'un côté, et *module* et *algorithme* de l'autre.

Fort de ces trois propositions nous nous positionnons dans le champ théorique de l'analyse. Nous soulignons une spécificité de la musique d'informatique, son écriture, qui ne nous paraît pas avoir été considérée dans les théories analytiques. En effet nous montrons que la réflexion analytique sur ces musiques s'est essentiellement faite à partir d'une absence d'écriture. Notre positionnement reprend alors celui présenté par [Clarke, 2006], où l'analyse s'effectue au sein d'un jeu de renvois entre l'écoute et la saisie de l'artefact.

Le second temps de notre thèse est dédié à la mise en pratique de nos propositions sur *Jupiter*. Cette pièce est particulière puisqu'elle intègre une possibilité de routing entre les modules, mais sans système de représentations viables. Pour y remédier nous avons développé

un programme *ad hoc* qui construit des représentations à partir des nombreuses écritures informatiques (les *qlist*). Ensuite nous effectuons le travail d'analyse dans le temps de la pièce, section par section. La saisie de l'artefact nous permet de démanteler les deux algorithmes de la pièce, puis de les resituer au sein d'une *musique interactive*. Malgré les nombreuses analyses effectuées sur cette pièce notre approche nous montre qu'une saisie de l'artefact permet d'accéder à une certaine profondeur, d'une façon qui ne peut être comparée à ce qu'autorise une analyse aurale, ou encore avec la prise en compte d'un fragment de l'artefact. Notre analyse permet par exemple de relever des éléments saillants présents dans la poïétique et qui pourtant restent absents de l'esthétique.

Abstract

This thesis is about the medium of computer music in a perspective of musical analysis. The specificity of this digital medium is used for questioning the practice of musical analysis. After the key works of [Battier, 2003] and [Manning, 2006], and moreover after having noticed that this music exists with those mediums, we make the hypothesis that those mediums are a rich resource for musical analysis. Then, our problematic is "how to comprehend the music that those mediums compute?". We build an answer in two parts. At first we expose some theoretical tools, and secondly we apply them in analyzing the musical piece *Jupiter* (1987) by Philippe Manoury, for flute, treatments and electronic.

Our theoretical tools are preceded by some terminological reflections. Quickly it appeared that what we are studying is not the *patch*, nor the *code* or the *sketch* but much more the musical ideas, which contributed to the code during the composition process. Hence instead of using the terms code or sketch we make the choice to use the term artifact. We use this term first to enlighten the heuristic power of the medium, and then to focus on the musical ideas that the code crystallizes.

We answer our problematic through three propositions. The first one is to conceive a system including Man, Artifact and Sound. Indeed it is in considering the artifact inside a system, within which a large number of interactions take place, that the musical ideas can be reached. Hence, our second proposition is to conceive the composition process as a highly dynamic process. We enlighten this process by using Charles Sanders Peirce's semiotic, which enables us to identify innumerable meanings. In a same way we understand that the work of musical analysis also has to take place within the same process, from where the meanings spring up. Our last proposition is to define a conceptual framework which allows us to describe the artifact. This framework is based essentially on the definition of variables, parameters and control on the one side, and Modules and Algorithms on the other.

With this three propositions we position ourselves in the scope of musical analysis. We first emphasize the fact that computer music has its own writing system, which seems absent in the theoretical analysis. Indeed we show that the reflection in the analysis of computer music is based on the absence of writing. Our position resumes that of [Clarke, 2006], where the analysis is built with a convergence between listening and the knowledge we get from understanding the artifact.

The second part of our thesis is dedicated to analysis. We apply our last theoretical proposition to the analysis of *Jupiter*. This piece is particular because it includes a possibility for writing the internal routing but without viable representation. We find a solution by building a program which scans all digital writing and builds some exhaustive representation so that we can know who produces the sound we are listening to. Then we analyze the work section by

section. The understanding of the artifact allows us to enlighten the two algorithms and then to describe them in the context of interactive music. Even if there are many analyses concerning Jupiter, our approach shows that the understanding of the artifact allows us to make a deeper analysis than an aural analysis would, or even an analysis which uses a fragment of the artifact. Our analysis shows for example that there are some key elements present in the poïetic that are absent from the eshtetic.

Remerciements

Je souhaite remercier tout d'abord mes directeurs, Kevin Dahan et Martin Laliberté. Mon projet de recherche a été accepté avec enthousiasme et ils ont su l'accompagner dans de bonnes conditions jusqu'à son accomplissement. Leur complémentarité m'a été d'une aide précieuse, particulièrement pour l'approche globale que j'adopte le long de cette thèse.

Ma reconnaissance va ensuite envers Philippe Manoury pour les échanges écrits comme oraux que nous avons eus à plusieurs reprises. À l'aune des 30 ans de *Jupiter*, Philippe Manoury a accepté avec un vif intérêt de replonger dans la composition de cette pièce. Il a pris le temps nécessaire pour apporter les réponses à mes questions, qu'il en soit remercié. Dans une même lignée je remercie Miller Puckette pour les échanges que nous avons pu avoir, notamment pour ses retours sur les résultats que nous avons obtenus.

Je remercie aussi vivement Alain Bonardi pour les échanges que nous avons eus à propos de l'analyse des « musiques interactives ». Mes travaux ont été accueillis avec enthousiasme et j'ai pu les prolonger dans un séminaire doctoral idoine.

Cette thèse a été réalisée en travaillant partiellement à la bibliothèque universitaire de l'UPEM. Nous souhaitons remercier son personnel pour les opportunités dont j'ai pu bénéficier.

Je remercie aussi mes amis qui ont su me supporter, surtout cette dernière année, Alexandre, Aurélien, Natacha, Gérald, Raymi, ceux habitués à se regrouper autour d'Émile Bataille, et ceux des montagnes.

Enfin je remercie ma famille, mes parents pour leur soutien dans le chemin parcouru, je leur dédie cette thèse.

Introduction

Notre recherche questionne l'analyse des musiques d'informatique. Le terme *musique d'informatique*, emprunté à [Dufourt, 2004], fait ici référence à des musiques où la composition s'est faite avec des langages de programmation, de sorte que ce qui est perçu résulte de calculs audionumériques.

Nous proposons d'introduire notre sujet en deux temps, d'abord à l'aide de la célèbre tripartition sémiologique esséminée par J.-J. Nattiez, et ensuite au travers d'une réflexion sur les signes graphiques de la partition et l'analyse musicale.

Dans [Nattiez, 1987, p. 37] l'auteur envisage la tripartition sémiologique comme l'articulation de deux processus avec un objet central : les processus poïétique et esthésique s'articulent avec la partition traditionnelle. Si nous appliquons ce schéma aux musiques d'informatique nous constatons aisément que ce qui occupe la place de la partition est maintenant un vaste ensemble de fichiers, des fichiers de données, des fichiers audios et d'autre qui se rattachent à des langages de programmation. Cet ensemble de fichiers – qu'ils contiennent des lignes alphanumériques ou des graphiques – est en effet bien conçu dans le temps de la composition et peut, par conséquent, être envisagé comme des traces du poïétique (e.g. [Zattra, 2015]). De l'autre côté de la tripartition nous constatons que ce qui rend sonore la pièce musicale, ce qui initie le processus esthésique, est aussi cet ensemble de fichiers. En effet ce sont bien ces derniers qui vont calculer les signaux audionumériques de la pièce. Il n'est donc pas incohérent de relever qu'en musique d'informatique, l'articulation des deux processus clés de la tripartition sémiologique, s'effectue par l'existence de fichiers informatiques. Alors, à la lumière de l'importance de la partition traditionnelle pour l'analyse musicale, notre recherche fait l'hypothèse que ces fichiers informatiques sont une ressource conséquente pour l'analyse musicale.

Le second moyen d'introduire notre sujet s'imprègne aussi de la sémiologie mais ici à l'échelle des signes graphiques. La sémiologie nous montre qu'un signe graphique n'est pas créé *ex nihilo* mais, à l'inverse, qu'il correspond à une appréhension singulière des choses. Un signe graphique renvoie ainsi à quelque chose d'autre que lui-même¹. Parallèlement, des lectures de M.E. Duchez sur la genèse de l'écriture musicale entre le IXe et le XIe siècle – essentiellement [Duchez, 1982] – nous ont montré que les signes graphiques de la partition en cette époque étaient moins à éclairer par le sonore que par la théorie musicale occidentale. Nous relevons ainsi que le signe graphique de la partition, ou plus justement le système de signes, correspond à une appréhension singulière du musical.

1 « Un signe est une chose qui, outre l'espèce ingérée par les sens, fait venir d'elle-même à la pensée quelque autre chose » Saint Augustin, cité dans [Barthes, 1964, p. 103].

Alors, quelque peu hâtivement, nous pouvons comprendre l'analyse musicale comme une pratique qui se réalise, notamment, avec les différentes appréhensions musicales sous-tendues par les signes de la partition. De façon imagée une analyse musicale peut être vue comme un chemin construit à l'aide de différents éléments, dont certains sont puisés dans ces appréhensions musicales auxquelles correspondent les signes graphiques de la partition.

Par analogie, nous pouvons d'abord comprendre que les signes présents dans les fichiers informatiques – ou plutôt des groupements de lignes de code – correspondent à des appréhensions. Pensons par exemple à l'implémentation d'une méthode de synthèse, qui correspond à une saisie sur le sonore (*e.g.* les formants de la synthèse Phase Aligned Formant). Ensuite, notre travail peut s'éclairer à l'aide de la précédente image du travail d'analyse musicale. Notre recherche peut s'envisager comme un moyen pour permettre à l'analyste de tracer son propre chemin, en puisant dans les différentes appréhensions reflétées par les contenus des fichiers.

Notre sujet se situe ainsi à la croisée de l'analyse musicale et des musiques d'informatique. Il vient du constat que le médium de ces musiques ne peut se saisir de la même façon que celui avec lequel l'analyse musicale s'est essentiellement développée. Avant de présenter notre problématique il nous est nécessaire d'éclairer le terme clé de notre recherche, *artefact*.

Notre compréhension de ce médium s'est effectuée à l'aide d'écrits en intelligence artificielle autour des années 1980. Par exemple, dans la préface du célèbre *Structure and Interpretation of Computer Programs*, nous pouvons lire « computer language is not just a way of getting a computer to perform operations but rather that it is a novel formal medium for expressing ideas about methodology » [Abelson *et al.*, 1984]. Cette compréhension sera reprise dans le champ musical par le chercheur compositeur Otto Laske. Pour ce dernier, dans l'acte compositionnel, l'ordinateur fait figure *d'alter ego* et, le programme réalisé – ce que nous avons désigné par l'ensemble de fichiers – est envisagé comme une objectivation de la pensée [Laske, 1990]. Dès lors, les outils de la musique d'informatique « permit one to trace one's process, observe cause and effect relationships between one's process and the resulting music » [Laske, 1992].

Ces idées nous ont amenés à dépasser une vue « centrée sur le sonore ». Dans les mêmes années, chez J.-C. Risset, les fichiers informatiques des pièces musicales – où plus justement les lignes alphanumériques contenues – sont envisagés comme des descriptions de la structure physique des sons (*e.g.* [Risset, 1991]). Or, dans une entreprise visant à éclairer la musique, il est aussi nécessaire de viser les idées qui ont régi la conception de ces lignes de code.

Nous avons ainsi été confrontés à un problème terminologique. Les termes *fichier informatique* et *medium*, utilisés dans cette introduction, restent trop imprécis. Dans la littérature scientifique ces derniers sont remplacés par *code*, *patch* ou encore *sketch*. Mais leur emploi nous pose problème. En effet, un travail qui étudie le *code* (ou *patch* ou *sketch*) pour éclairer la musique reste problématique, puisque ce qui est visé se trouve en amont du code. Il ne s'agit pas du code, mais des idées avec lesquelles celui-ci s'est formé ; le code doit être dépassé pour atteindre des idées musicales.

Ce problème terminologique s'est résolu de façon empirique avec des excursions dans la genèse de l'écriture de la musique occidentale. Notre intérêt pour cette période se trouve essentiellement dans la richesse du signe graphique pour la cognition de l'Homme, le pouvoir heuristique de la notation musicale. Alors, en concevant qu'un langage de programmation est aussi une heuristique pour la composition, nous avons effectué des analogies entre une écriture graphique et une computationnelle. À la suite d'une multitude de renvois, par jeu de miroirs, et avec le célèbre article *L'artefice d'écriture* de Hugues Dufourt en toile de fond, c'est le terme *artefact* – du latin *ars* (art) et *facere* (faire) – qui nous a semblé le plus approprié.

Dans le champ de l'informatique, nous entendons par *artefact* un ensemble d'idées véhiculées par du code. L'artefact est implémenté avec du code, ce qui détermine son apparence. Il a ceci de particulier qu'il peut être implémenté dans un autre code que celui original, et que la lecture de cette nouvelle apparence par une tierce personne permet aussi d'accéder au même artefact, et ainsi de suite. L'artefact est donc abstrait. Il se saisit par une surface lisible, faite de signes alphanumériques, une implémentation, et il peut avoir plusieurs de ces surfaces².

C'est à la lumière de ce mot clé *artefact* que se dessine notre problématique. En considérant qu'ils intègrent des idées musicales nous nous posons la question suivante : « comment saisir ces artefacts pour développer des analyses musicales ? »

2 Il est possible que l'association de ces fichiers informatiques à des artefacts pose problème pour certains lecteurs. En effet un usage courant du terme artefact en archéologie nous amène à l'associer à quelque-chose de matériel, qui résiste au temps. L'usage que nous faisons de ce terme correspond au sens général hérité de l'archéologie par le fait que dans un cas comme dans l'autre ces « objets » sont considérés comme des ressources pour étudier une pratique humaine. Ensuite, notre usage s'en éloigne évidemment sur l'aspect matériel : un artefact de musique d'informatique n'a pas la même matérialité que celle d'un vase de poterie. Ces analogies nous amènent à des questions profondes sur l'immatérialité des objets de notre époque numérique. Cette immatérialité ne doit pas cependant nous empêcher de les considérer comme des richesses pour « l'étude de l'Homme ». Par exemple en 2017 l'INRIA a mis au jour un projet, *Software Heritage*, qui collecte, préserve et rend accessible les *codes sources* d'un très grand nombre de logiciels (plus de 55 millions en avril), lesquels sont envisagés comme « la composante principale de notre patrimoine scientifique, technique et culturel » (cf. [Di Cosmos, 2017]). En utilisant le terme *artefact* nous souhaitons souligner la richesse de ces objets numériques et par la même éviter une circonscription de ces derniers aux sciences informatiques ou mathématiques. Nous ne pouvons pas ici davantage justifier l'emploi de ce terme. Un travail intéressant et conséquent reste à faire sur ce sujet, moins sur l'aspect technologique de la pérennisation que sur la façon d'envisager ces objets dans notre discipline.

Notre recherche s'articule en trois parties. La première, nommée *approches*, met en avant les caractéristiques des artefacts. À l'aide d'exemples d'implémentation pédagogique nous avançons d'abord 5 points de singularité des artefacts. Ensuite, après avoir mis au clair un système clé avec lequel l'artefact prend sens, le système Homme-Artefact-Son, nous exposons les outils habituels qui permettent de saisir les artefacts, *e.g.* les dimensions discret/continu et signal/symbole.

Dans la deuxième partie nous avançons nos propres outils pour saisir les artefacts. C'est dans celle-ci, nommées *propositions*, que nous répondons à notre problématique. Après avoir apporté de la lumière sur la notion de matériau musical nous utilisons la sémiotique pour investir les différents niveaux d'abstraction présents dans les artefacts. Nous proposons alors un cadre conceptuel afin de les décrire, à partir essentiellement de leur structuration en niveaux d'abstraction. Enfin, après avoir constaté que la musique d'informatique possède une écriture, nous terminons cette partie en nous positionnant dans le champ théorique de l'analyse musicale.

La dernière partie, *application*, est une mise en pratique de nos propositions. Nous analysons une pièce célèbre du répertoire, *Jupiter* (1987), composé par Philippe Manoury à l'Ircam dans les années 1980 avec Miller Puckette, pour flûte, traitements et électronique. Dans un premier temps nous décrivons l'artefact de la pièce à l'aide du précédent cadre théorique ; dans un second nous confrontons nos connaissances de l'artefact à l'écoute de la pièce afin de l'analyser section par section.

La pièce *Jupiter* a été choisi dès le début de nos travaux. Très tôt nous avons eu un intérêt pour la « musique en temps réel » [Manoury, 2007] qui, en plus d'associer instrument traditionnel et production électronique, tisse des liens techniques et esthétiques entre ces derniers. Aussi, dans la même lignée, nous avons eu un intérêt pour des musiques où l'effectif permet d'exploiter cette rencontre, c'est-à-dire un effectif simple, entre un instrumentiste et l'électronique. C'est donc assez naturellement que les pièces de P. Manoury se sont imposées. *Jupiter* s'est ensuite démarquée assez vite puisqu'il s'agit d'une pièce pionnière : elle ouvre en effet le cycle *Sonus Ex Machina*, avec lequel le compositeur et son équipe ont apportés les premiers outils techniques et les premières réflexions théoriques et esthétiques sur cette musique. Aussi cette pièce restait, selon nous, moins étudiée que d'autre comme *Pluton* (1988, pour piano et électronique) ou encore *En Echo* (1994). Enfin, son artefact (ou *patch*) a été mis à disposition sur internet dans le cadre d'un projet aussi novateur qu'ambitieux, « New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics », [Puckette,

2001]. C'est pour l'ensemble de ces raisons que nous avons choisi d'appliquer nos propositions théoriques à *Jupiter*.

Notre problématique néanmoins ne questionne pas la période historique dans laquelle *Jupiter* s'inscrit. En effet nous n'avons pas voulu restreindre la question de l'analyse à des musiques d'une certaine période – qui par exemple auraient en commun l'usage d'une même technologie (e.g. la $4X$). Notre problématique reste donc générale et questionne l'analyse musicale des pièces d'informatique à partir de leur artefact. Il nous a en effet semblé que la délimiter historiquement aurait pu limiter le champ de réflexions. En contrepartie il va de soi que face à un questionnement si ouvert une thèse ne peut être qu'un apport. C'est dans cette perspective qu'il faut envisager *Jupiter*; c'est avec cette pièce pionnière que nous avons choisi de construire notre apport. Les travaux fondateurs de ce questionnement sont ceux de A. Di Scipio sur l'informatique et la musique électroacoustique [Di Scipio, 1995a] ; ceux de M. Battier sur la *faktura* [Battier, 2003] et enfin ceux de P. Manning sur la *techné* [Manning, 2006].

Première partie

Approche

Chapitre I

Points de singularités des artefacts

Résumé

Ce chapitre est une première approche des artefacts dans le but de dégager des points de singularités. Elle vient faire face au fait que les artefacts sont sans précédent dans l'ensemble des « traces du poïétique ». L'approche permet ainsi de situer différentes spécificités, ce qui doit permettre d'éviter de se focaliser sur l'une seule d'entre elles et ainsi de permettre une approche globale. Cinq points de singularités sont avancés : les artefacts (i) peuvent posséder plusieurs apparences, (ii) cristallisent des connaissances qui dépassent le musical, (iii) sont le fruit d'une activité de *design*, (iv) possèdent une sémantique et (v) sont faits d'une densité colossale.

Ces points sont avancés à partir d'exemple d'artefact. Dans le premier grand titre nous utilisons la célèbre simulation de cloche de J.-C. Risset ; dans le second un fragment d'artefact utilisé pour sa pièce *Inharmonique* (1977) pour soprano et bande.

1 Points de singularités

Afin d'appuyer notre réflexion nous donnons en liminaire un exemple d'artefact. Nous utilisons ici un exemple pédagogique, devenu un classique de l'histoire de l'informatique musicale : la simulation des sons de cloche par Jean-Claude Risset. L'importance de cet exemple est telle qu'on le retrouve dans différents textes majeurs de l'informatique musicale, nous en utilisons trois : [Risset, 1969], [Dodge, 1997] et [Puckette, 2006].

1.1 Exemple d'artefact pédagogique

1.1.1 Jean-Claude Risset et Music V

En 1969 Jean-Claude Risset, compositeur chercheur et pionnier de l'informatique musicale, sur lequel nous reviendrons tout au long de ce premier chapitre, publie un document qui fera date, nommé « Catalog of computer-synthesized sounds » ([Risset, 1969]). Ce catalogue est composé d'une trentaine d'exemples de synthèse sonore, qui simulent des sons essentiellement instrumentaux. Nous utilisons ici l'exemple des sons de cloches (numéro 430, p. 202), qui sera repris plus tard – ce qui contribuera à le rendre célèbre – dans une des compositions de l'auteur, *Inharmonique* (1977) pour soprano et bande.

```
COMMENT-----JCR430----- ;
COMMENT : BELL experiemnts ;
COMMENT : ON TAPE M1485 FILE 4 ; GEN 0 5 3 ;
COMMENT: 5 KC SAMPLING RATE . SIA 0 5 5000 ;
INS 0 1 ; OSC P5 P7 B3 F2 P30 ; OSC B3 P6 B3 F1 P29 ;
OUT B3 B1 ; END ;
COMMENT : TO SET GENERAL CONVT ; SV2 0 10 2 6 -7 ;
COMMENT : SYNCHRONOUS DECAY ;
NOT 1 1 20 250 224.5 ; NOT 1 1 20 400 368.5 20 ;
NOT 1 1 20 400 476 20 ; NOT 1 1 20 250 684 20 ;
NOT 20 220 800 20 ; NOT 1 1 20 200 1096 20 ;
NOT 1 1 20 200 1628 20 ;
SEC 21 ;
```

Comme le rappelle la deuxième ligne, l'exemple permet de simuler, à l'aide du langage Music V, sur lequel nous reviendrons plus tard, un son de cloche. En plus de ce code deux informations textuelles sont ajoutées dans le document :

- F_2 est une table qui décroît logarithmiquement de 1 à 2^{-10}
- F_1 est une fonction sinusoïdale

1.1.2 Charles Dodge et la théorie

Quinze années après la parution du catalogue de J.-C. Risset (1985), le compositeur Charles Dodge reprend cet exemple dans un ouvrage qui fera aussi date, [Dodge, 1997]³. À la différence, la simulation de cloche par synthèse n'est pas présentée par l'intermédiaire d'un code informatique mais à l'aide de trois éléments : un schéma de synthèse, une table de données, et moins d'une vingtaine de lignes de texte (cf. p. 105). La table donne pour chacun des partiels un triplet de données (amplitude, durée et fréquence) en valeurs ordinales, par rapport au premier partiel.

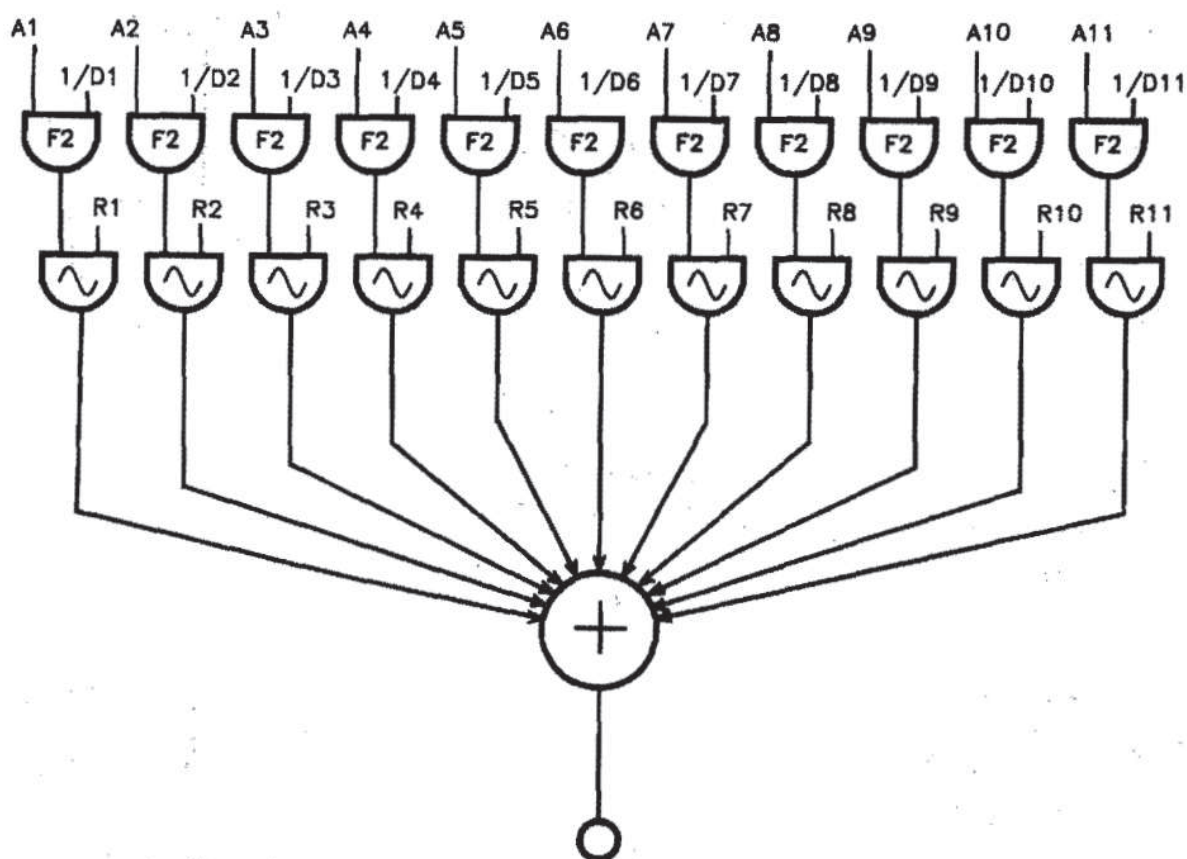


Fig. 1 - Schéma de synthèse pour simuler des sons de cloches, extrait de [Dodge, 1997]

3 L'ouvrage est paru en 1985, suivi d'une seconde édition en 1997.

Amplitudes		Durations		Frequencies	
A1	AMP	D1	DUR	R1	FREQ*.56
A2	AMP*.67	D2	DUR*.9	R2	FREQ*.56+1
A3	AMP	D3	DUR*.65	R3	FREQ*.92
A4	AMP*1.8	D4	DUR*.55	R4	FREQ*.92+1.7
A5	AMP*2.67	D5	DUR*.325	R5	FREQ*1.19
A6	AMP*1.67	D6	DUR*.35	R6	FREQ*1.7
A7	AMP*1.46	D7	DUR*.25	R7	FREQ*2.
A8	AMP*1.33	D8	DUR*.2	R8	FREQ*2.74
A9	AMP*1.33	D9	DUR*.15	R9	FREQ*3
A10	AMP	D10	DUR*.1	R10	FREQ*3.76
A11	AMP*1.33	D11	DUR*.075	R11	FREQ*4.07

Fig. 12 - table de donnée pour simuler des sons de cloches,
extrait de [Dodge, 1997]

1.1.3 Miller Puckette et Pure Data

Quasi une décennie après, Miller Puckette, chercheur en informatique musicale, reprend le précédent exemple de Charles Dodge dans un nouveau langage de programmation conçu par lui-même, Pure Data (cf. [Puckette, 2006]). L'exemple permet à M. Puckette de présenter un des principes clés de son langage, des *abstractions*. Ce principe lui permet de structurer, et par là même d'alléger, son artefact en deux niveaux :

- un premier qui concerne la génération des partiels : une sinusoïde à une certaine fréquence, avec une enveloppe d'amplitude percussive et quadratique, durant un certain laps de temps. Ceci est réalisé par une abstraction idoine, nommée *partial*. (cf. Fig. 3)
- un second qui concerne l'actualisation et la sommation de chacun des partiels, où des valeurs d'amplitude, de durée, de fréquence et enfin de « désaccordage » (*detune*) sont précisées en fonction de valeurs de référence, par l'intermédiaire de coefficients (cf. Fig. 4). Ces coefficients sont entièrement repris de la table de données de C. Dodge. (cf. Fig. précédente).

Avec ce procédé M. Puckette réimplémente en fait l'idée riche de C. Dodge, celle de déduire les caractéristiques des partiels en fonction de valeurs de référence.

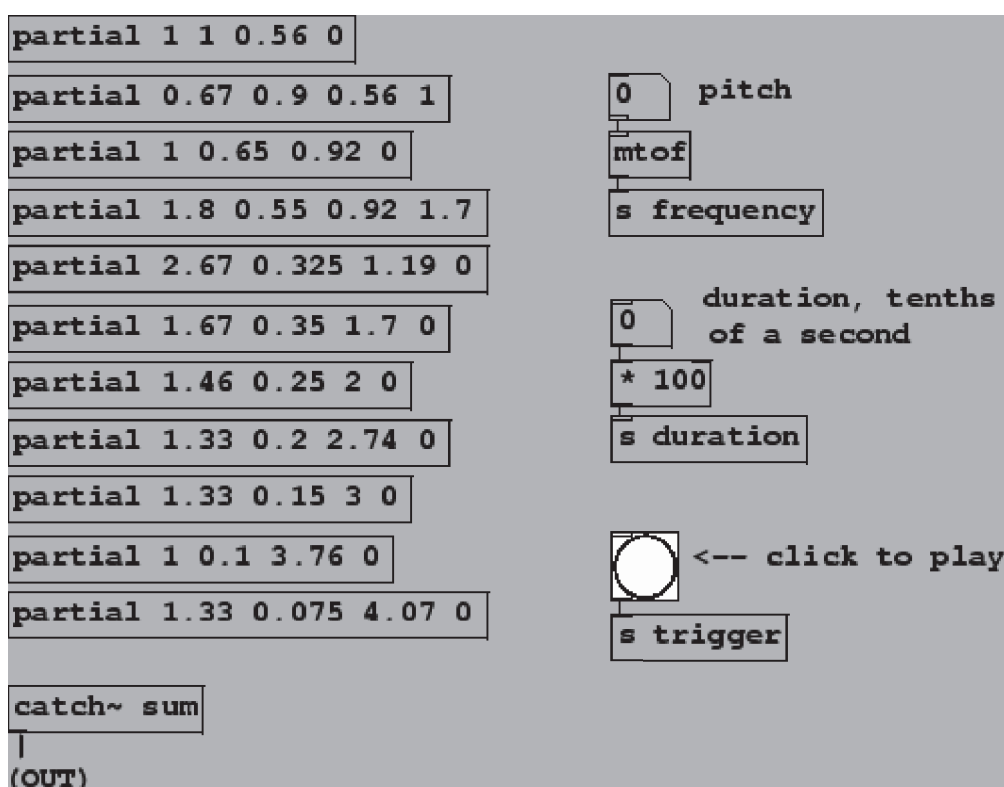


Fig.3 - Patch Pure Data, simulation des sons de cloche, niveau global, capturé de [Puckette, 2006]

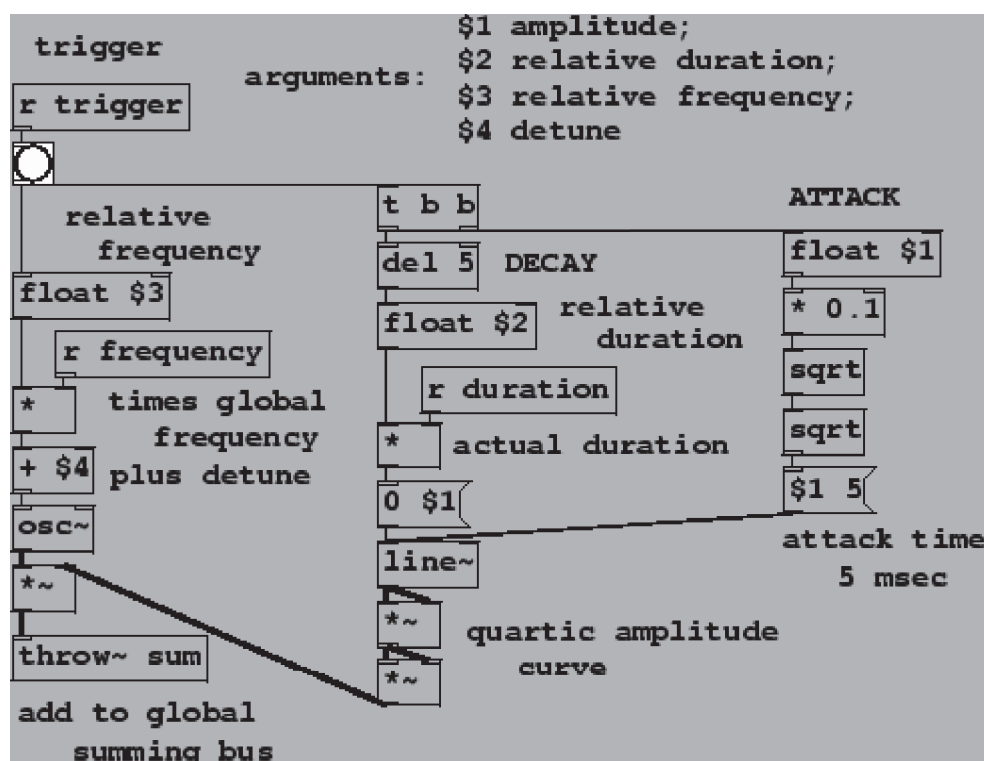


Fig.4 - Patch Pure Data, simulation des sons de cloches, niveau local : définition de l'abstraction partial capturé de [Puckette, 2006]

1.2 Singularité : des apparences

Le premier point de singularité est peut-être le plus difficile à cerner. Il vient du fait qu'un artefact peut avoir plusieurs *apparences*. C'est ce que nous montrent les exemples de J.-C. Risset et de M. Puckette : un même son de cloche, selon le langage utilisé, peut avoir plusieurs apparences. Ainsi, par extrapolation, nous déduisons que l'artefact d'une pièce d'informatique musicale peut avoir plusieurs apparences. C'est-à-dire qu'en tant qu'analyste, l'étude d'une pièce à partir de son artefact peut se faire avec des « fichiers informatiques » qui n'ont entre eux, si l'on s'arrête aux lignes qui les structurent, que peu voire pas de choses communes.

Dans notre entreprise visant en premier lieu à explorer les artefacts, il nous faut alors instaurer une relation particulière entre l'artefact et ses multiples apparences possibles, une relation que nous qualifions en conséquent de *one to many* : ce que nous nous pouvons lire, étudier, communiquer, publier, comprendre, peut avoir plusieurs apparences, alors qu'ils traitent tous « d'une même chose ».

Ceci se retrouve de façon flagrante dans la jeune histoire de la musique d'informatique avec le problème maintenant connu de l'obsolescence technologique : des pièces de moins d'un demi-siècle peuvent ne plus être actualisées du simple fait que le matériel qu'elles sollicitaient est devenu, avec l'inexorable développement technologique, obsolète. Citons par exemple la pièce historique *Stria* (1977) de John Chowning, pour bande : composée dans les années 70 avec le langage historique Music V, elle fut reconstruite dans les années 2000 avec un langage plus récent, hérité du précédent, Csound (cf. [Dahan, 2007a] par exemple).

Dans le domaine informatique, cette relation particulière entre l'artefact et ses apparences, a pour corrélat le terme d'*implémentation*. Par exemple en ingénierie, quand une solution est trouvée, elle est ensuite *implémentée* dans des technologies afin d'être rendue effective. Et ces technologies résultent de choix : la solution aurait pu être implémentée dans d'autres technologies. Ainsi, quand [Dahan, 2007a] fait le choix d'implémenter une partie de l'artefact *Stria* dans le langage Ruby, [Baudoin, 2007] choisit lui un autre langage, nommé Python.

Si l'artefact a un but, disons que celui-ci peut être atteint par plusieurs chemins, et que chacun de ceux-ci engendre une apparence particulière de l'artefact⁴.

1.3 Singularité : des connaissances

Le second point de singularité concerne les connaissances présentes dans l'artefact. Pour les investir nous proposons de différencier trois types, non exclusifs les uns des autres. Des

4 Ici apparaît une difficulté dans la conservation des pièces et de leurs artefacts : chacun de ces chemins est tributaire de l'environnement technologique d'une époque.

connaissances (i) de conventions formelles, (ii) de synthèses et de traitements des signaux, et (iii) de « recettes de fabrication ».

1.3.1 Conventions formelles

La première découle du fait que l'artefact est implémenté. Pour réaliser cette implémentation il est nécessaire de connaître (au moins) un langage de programmation, soit, pour reprendre [Risset, 1996], de maîtriser des *conventions formelles*.

Notons que l'usage de langages de programmation, la maîtrise de ces conventions formelles, demandent un temps d'apprentissage qui ne peut être négligé.

Par exemple, la première simulation de cloche, extraite de [Risset, 1969] nécessite la connaissance du langage Music V ; la dernière, extraite de [Puckette, 2006], nécessite, elle, la connaissance de Pure Data.

1.3.2 Synthèses et traitements de signaux

Le second type de connaissance est relatif à la génération de signaux audio. Ce qui n'est pas dit dans l'exemple historique de J.-C. Risset, c'est que le son de cloche est simulé avec une méthode de synthèse qui consiste à additionner des sinusoïdes, la synthèse additive. À notre époque d'autres méthodes pourraient être utilisées pour effectuer cette simulation, comme la synthèse par modulation de fréquence par exemple. Il existe en effet un ensemble de méthodes de synthèse, et, parallèlement, il existe aussi un ensemble de méthodes pour modifier des signaux déjà existants. Ces derniers sont en lien avec un domaine de l'ingénierie, le traitement de signaux numériques (Digital Signal Processing : DSP).

Si l'artefact a pour fin de produire un signal numérique, il est donc nécessaire, pour le « construire », de maîtriser des méthodes de synthèse et de traitement de ces signaux.

Ce type de connaissance s'illustre parfaitement quand la création d'une pièce est concomitante à une nouvelle méthode de synthèse ou traitement. C'est par exemple le cas pour *Stria*, qui correspond à la première utilisation de la synthèse par modulation de fréquence, inventée et brevetée par le compositeur même (cf. [Baudoin, 2009]). C'est aussi le cas pour la pièce que nous analyserons, *Jupiter* (1987) de Philippe Manoury, qui a recours, après sa création, à la méthode de synthèse « Phase Aligned Formant » (P.A.F.), inventée par M. Puckette au début des années 1990 (cf. [Puckette, 1996])⁵.

5 La synthèse PAF fut ajoutée quasi dix ans après la création de la pièce, en 1996, ce qui engendre une nouvelle version de la pièce. (cf. Chap VI 1.2).

1.3.3 Recettes de fabrication

Le dernier type de connaissance est le moins évident des trois. Il est présent dans le texte accompagnant l'exemple théorique de C. Dodge :

The three principal features that contribute to the bell-sound-like sound are (1) non harmonic partials ; (2) decay times of the partials roughly inversely proportional to their frequency and (3) beating of pairs of components, slightly mistuned on the lowest two partials. [Dodge, 1997, p. 104]

Si le dernier le point (3), concernant les battements, peut être associé au domaine du DSP, il n'en va pas de même des deux précédents : ils s'apparentent davantage à des *connaissances* qui sont, en amont, connues des programmeurs, et en aval, cristallisées dans l'artefact.

C'est ici que réside une particularité du « catalogue de Risset » – qui vient aussi expliquer son rayonnement : en plus d'intégrer des implémentations, des nouvelles connaissances sont présentes, qui renseignent ici sur la structure du son. Avec l'auteur nous nommons ces dernières des « recettes de fabrication » :

[the computer score and some additional explanations presented in the catalog] give the recipe for synthesis and also affords a through description of the physical structure of the sounds [Risset, 1969, p. 112]

Il est facile de communiquer les recettes de production des sons, sous forme des prescriptions qu'on fournit à un programme de synthèse donné comme Music V [Risset, 1992, p. 6]

Ainsi, en filant la métaphore, on peut résumer la précédente recette de la façon suivante : afin de simuler des sons de cloche avec la synthèse additive, il est nécessaire (i) de produire 9 partiels en respectant la distribution fréquentielle $f_n/f_0 = 0.5, 1, 1.2, 1.5, 2$ (avec n le rang du partiel ; f_n la fréquence du partiel de rang n ; f_0 la fréquence fondamentale), (ii) d'appliquer à chacun des partiels une enveloppe d'amplitude percussive logarithmique dont (iii) la durée est inversement proportionnelle à leurs fréquences. Enfin, pour plus de réalisme (iv) les deux premiers partiels peuvent être remplacés par deux paires de partiels de sorte à provoquer des battements.

Dans les exemples du catalogue de Risset ces recettes sont issues de simulations de l'environnement musical traditionnel, et concernent, de fait – comme cela est précisé dans l'introduction (p. 113) –, la psychoacoustique. Mais comme l'a montré l'auteur avec sa pièce *Inharmonique*, ces « recettes » peuvent déborder de la simulation pour être davantage en lien avec un processus compositionnel.

1.4 Singularité : du design

Le troisième point de singularité des artefacts est qu'ils sont le fruit d'une activité de *design*. Nous pouvons illustrer ce terme par exemple avec l'implémentation de M. Puckette. Celle-ci nous permet en effet d'observer des choix réalisés par le concepteur :

1. l'amplitude générale est figée ;
2. l'enveloppe d'amplitude des partiels a un temps d'attaque de 5 ms ;
3. l'enveloppe d'amplitude des partiels possède une forme quadratique ;
4. la hauteur du son de cloche se prescrit en valeur MIDI ;
5. la durée est prescrite en 100^{ème} de seconde ;
6. une abstraction Pure Data permet de générer les partiels.

Relevons que les choix 3 et 4 sont issus de connaissance en psychoacoustique, et que le choix 6 est une technique d'implémentation, propre au langage utilisé.

Le terme *design* peut alors s'éclairer en constatant que le travail réalisé réunit d'une part des connaissances psychoacoustiques et techniques, et d'autre part des choix personnels (1, 2 et 5). On peut alors comprendre ce troisième point de singularité comme la confrontation des deux précédents : le *design* dans notre contexte, c'est l'utilisation de connaissances par une ou quelques personne(s) pour réaliser, *via* l'implémentation, quelque chose de nouveau⁶.

L'artefact n'a alors rien de commun, il est le fruit d'un (ou de) *designer(s)* qui possède(nt) ses (leurs) propres *idiosyncrasies*⁷.

Remarquons enfin que ces spécificités viennent redoubler le premier point sur les apparences possibles d'un artefact. Après avoir constaté que l'artefact peut avoir, en puissance, autant d'apparences qu'il y a de langages de programmation appropriés, il nous faut de plus ajouter que ces apparences varient selon les designers qui les réalisent.

6 « [design is] the interaction between understanding and creation » [Winograd *et al.*, 1994, p. 4].

7 À la différence de la notation de la partition traditionnelle, que l'on a qualifiée volontiers de *conventionnelle* [Seeger, 1958].

2 Points de singularités *bis*

Afin d'amener deux singularités supplémentaires nous présentons d'abord un exemple d'artefact. Il s'agit maintenant d'un fragment qui correspond à un extrait de la pièce *Inharmonique* de (1977) de J.-C. Risset, pour bande et soprano. Ce fragment a été diffusé dans un travail original, qui a fait date, réalisé par Denis Lorrain [Lorrain, 1980], sur lequel nous revenons amplement au Chapitre VI. Disons simplement ici que le code présenté dans le document – implémenté dans Music V – est issu de l'original réalisé par J.-C. Risset, et que le travail de D. Lorrain a consisté d'une part à le rendre plus cohérent, et d'autre part à ajouter d'importants commentaires qui viennent grandement faciliter la lecture⁸.

2.1 *Inharmonique*, section IV

Inharmonique se compose d'un total de 7 sections. Nous utilisons ici seulement la première séquence de la section IV, nommée BELHH₄. Pour présenter ce fragment d'artefact nous suivons la division classique du langage Music V, entre *score* d'une part, et *instrument* de l'autre. Cette division sera approfondie au prochain chapitre. Disons simplement que la première permet de prescrire le son souhaité, en termes de paramètres, tandis que le second permet de produire le signal sonore à diffuser.

2.1.1 L'instrument

Le fragment qui produit le signal est assez simple, il relie une enveloppe d'amplitude à un oscillateur sinusoïdal. « L'instrument » Music V peut ainsi être représenté :

8 Ce travail a ainsi été réalisé « à l'intention des stages Ircam pour compositeurs », précise J.-C. Risset dans la préface.

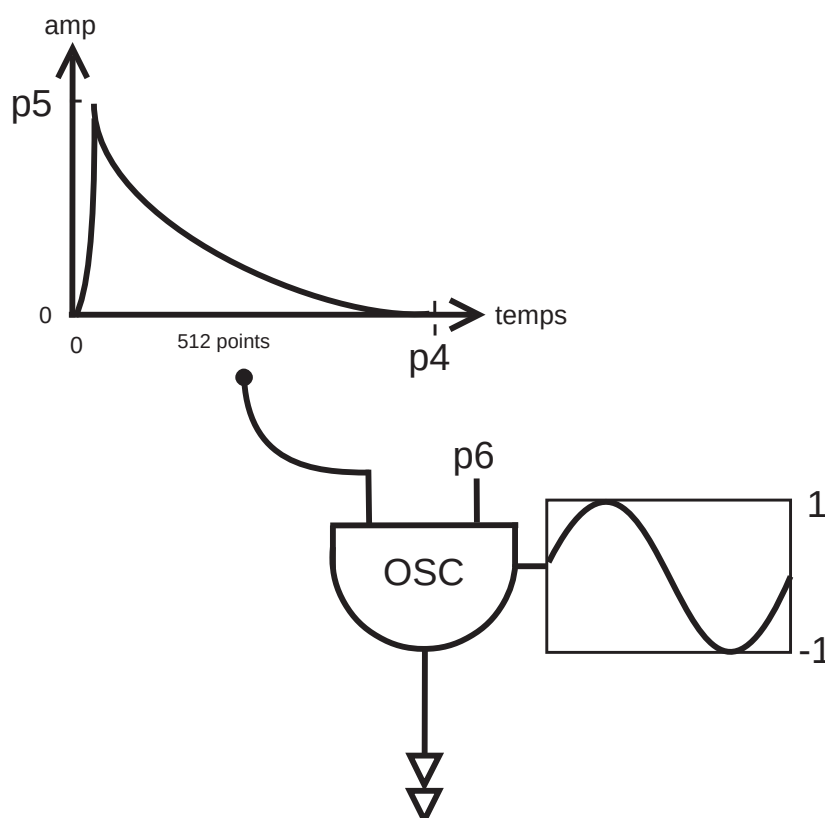


Fig.5: Instrument numéro 3 de la section IV d'Inharmonique

Il contient donc trois paramètres standard de synthèse, nommés P_4 , P_5 et P_6 , qui affectent respectivement la durée, l'amplitude et la fréquence du son à produire.

Cependant, l'implémentation réalisée va plus loin puisque le contrôle se fait par l'intermédiaire d'un élément qui se situe entre « l'instrument » et la « partition » du langage, appelé « structure de cloche » [Lorrain, 1980]. Celle-ci peut être comprise comme une structuration d'un ensemble de données qui permet de décrire un son de cloche. Les données précisées sont les suivantes : nombre de partiels, intensité globale, fréquence de référence, numéro de l'instrument puis un triplet de données pour chaque partiel (fréquence, durée, amplitude).

La première séquence de la section utilise une seule structure de cloche, numérotée 1280, avec les données présentées dans le tableau suivant :

Nombre partiels	Amplitude global	Fréquence de ref.	Num. instru	Triplet pour chaque partiel freq, durée, amplitude
9	975	349	3	675, 24, 200 124, 16, 200 346, 22, 200 1108, 17, 200 1634, 6, 200 2249, 3, 200 2941, 2, 200 3700, 1.5, 200 4504, 0.8, 200

Tab. 1 - Données de la structure de cloche numéro 1280 utilisées dans la section IV d'Inharmonique. Extrait depuis [Lorrain, 1980]

Nous comprenons maintenant l'utilité de cette structure : elle permet au compositeur d'engendrer des sons de cloches en précisant des paramètres globaux (durée, amplitude et hauteur), et non pas l'ensemble des partiels qui constituent le son⁹.

2.1.2 La « partition »

Le dernier fragment de code nécessaire pour cette séquence concerne les valeurs d'actualisation de cette cloche : comment la durée (P₄), l'amplitude (P₅), et la hauteur (P₆) sont-ils prescrits ?

Cette réalisation se fait, avec les yeux de notre époque, c'est-à-dire 40 ans après, de façon plutôt laborieuse. En plus de la structure de cloche, sont nécessités (i) une sous-routine (nommée PLF numéro 6), (ii) un instrument factice et (iii) une liste d'événements qui inclut les paramètres globaux d'amplitude et de hauteur.

Le code de la première séquence est le suivant. Il est fait d'un seul événement, ou « note » (NOT) :

```
INS 0 1 ; END ;
PLF 0 6 1 1280 ;
NOT 1 1 0 1000 440 ;
```

Ces trois lignes peuvent être expliquées sous forme littérale de la façon suivante :

- créer un instrument vide, numéroté 1 ;

⁹ Cette structure de cloche permet de pallier à la limitation la plus connue de la synthèse additive, la surcharge de paramètres impliqués.

- appeler la sous-routine numéro 6. L'appliquer au seul événement « note » qui lui succède, et le rediriger vers la structure de cloche numéro 1280 ;
- appliquer l'événement « note » au temps $t = 1$ sec, avec une durée nulle, avec les paramètres d'amplitude et de hauteur (P_5 et P_6) suivants 1000 et 440.

Afin de maîtriser l'ensemble de ce fragment, il nous reste à élucider la sous-routine PLF 6, qui fait le lien entre les événements « notes » et la structure de cloche. Elle est donnée par [Lorrain, 1980] sous forme littérale¹⁰. Comme il s'agit d'une forme conditionnelle, nous proposons de la résumer ainsi :

- Si le paramètre de durée (P_4) est nul, alors

$$\text{durée des partiels} = \text{duréeDuTriplet} * \frac{1}{P_6 / \text{freqRef}}$$

$$\text{durée des partiels} = \text{duréeDuTriplet} * \frac{\text{freqRef}}{\text{freqSouhaitée}}$$

- Sinon,
les durées sont ajustées proportionnellement de telle sorte que la composante la plus longue occupe la durée prescrite (P_4).

2.2 Singularité : sémantique

Le dernier point de singularité que nous souhaitons amener se trouve en substance dans les deux lignes de code suivantes. Celles qui vont « faire jouer » tel son de cloche à tel instant selon la volonté du compositeur.

```
PLF 0 6 1 1280 ;
NOT 1 1 0 1000 440 ;
```

Une fois mis de côté les éléments propres aux conventions du langage de programmation¹¹, il y a dans ces deux lignes quatre données essentielles, qui ont un impact sur le son engendré, soit avec lesquelles la composition va se faire. Il s'agit de 1280 et du 0, 1000 et 440 de la seconde ligne. Ce qui nous semble ici important c'est que ces données ne sont significatives qu'en connaissance des connexions qu'elles entretiennent, en connaissance de leurs destinations :

¹⁰ L'absence des PLF pose problème pour une simulation ou reconstruction de la pièce. [Sousa Dias, 2011] a travaillé avec le compositeur sur ces PLF, et les a ré implémenté avec Open Music, puis plus récemment JavaScript.

¹¹ Comme PLF, NOT, et les numéros d'instruments.

- 1280 fait référence à une structure de cloche ;
- 0, 1000 et 440 viennent paramétrer cette structure de cloche, en durée, amplitude, et en hauteur.

Ces données sont signifiantes pour le compositeur qui a intégré ces connexions lors de son travail de *design*. Mais pour toute autre personne, qui découvre le code une première fois, et même pour un expert, ces lignes s'avèrent *a priori* vides de sens. Pour l'analyste il est ainsi nécessaire de lire ce fragment d'artefact, de « suivre » les différentes connexions sous tendues par ces deux lignes, de sorte à intégrer leurs *significations*. Ensuite, dans la production de son travail d'analyse, il devra rendre compte de ces significations :

Les énoncés NOT eux-mêmes spécifient une amplitude (P₅) et une fréquence (P₆) : la structure sera transposée en amplitude et en fréquence d'après ces paramètres [Lorrain, 1980, section IV]

Dans la mesure où l'artefact intègre des significations qui résultent du travail si singulier de *design*, et dans la mesure où la composition s'est faite avec cet artefact sous les yeux, soit que les actions compositionnelles se sont faites avec l'intégration de ces significations, nous constatons que l'artefact intègre une *sémantique*.

Pour celui/celle qui n'est pas le *designer* de l'artefact, les lignes de code avec lesquelles la composition s'est faite restent à comprendre. Et cette compréhension passe par l'intégration des significations des éléments qui composent la ligne.

2.3 Singularité : densité

Le dernier point de singularité est lui plus trivial, il concerne la densité des artefacts. Le nombre de lignes de code présentes dans un artefact peut servir d'indicateur. Pour la pièce *Jupiter* (1987) nous avons ainsi relevé plus de 8 000 lignes dans un des constituants de l'artefact, que nous avons appelé la partition alphanumérique. En parallèle, pour ce qui concerne le « patch », nous pouvons citer le travail de [Dufeu, 2013], qui, pour la même pièce, a relevé plus de 14 000 boîtes objets Max¹². De façon similaire, pour la pièce *Partita I* (2006) pour alto et électronique du même compositeur, c'est cette fois plus de 40 000 objets Max¹³ qui ont été relevés dans le patch.

¹² « Le patch d'exécution de *Jupiter* (...) contient 14 290 objets Max, répartis dans 444 subpatchers » [Dufeu, 2013, paragraphe 3]. Il s'agit du patch nommé *Jupiter2007beta.pat* (cf. note 9).

¹³ « 40 111 objets Max, répartis dans 1 749 subpatchers », [Dufeu, 2013, paragraphe 3]. Il s'agit du patch nommé *PARTITA2011-24.maxpat* (cf. note 10)

Précisons dans notre optique que ces valeurs doivent être lues comme des *ordres de grandeur* et non avec précision. Ainsi la distinction entre « sous patch », « abstraction », « objet natif », « objet personnel » nous intéresse peu, car comme nous l'avons vu, les apparences des artefacts sont considérablement variables, d'une part selon le(s) langage(s) utilisé(s), et de l'autre selon le(s) designer(s).

Au regard de la partition traditionnelle cette densité des artefacts apparaît inévitablement comme exorbitante¹⁴. Elle nous amène finalement à rendre compte que l'activité qui consiste à lire et à comprendre les artefacts demande un temps qui ne peut être négligé.

Mais même pour qui est familier avec le langage, les partitions MUSIC V sont cryptiques, les processus y sont décrits de façon quasi-atomique : une explication peut aider à comprendre ce qu'ils visent. *Jean-Claude Risset dans la préface à [Lorrain, 1980]*

14 Pour pousser la comparaison, naïvement, si l'on comptait 40 lignes par page A4, l'impression de la partition alphanumérique de *Jupiter* nécessiterait 200 pages, quand la partition traditionnelle en nécessite 35.

Chapitre II

Systeme Homme Artefact Son et niveaux d'abstraction

Résumé

Ce chapitre poursuit la phase d'approche réalisée précédemment. Nous présentons d'abord un système avec lequel l'artefact s'inscrit inévitablement, le système Homme Artefact Son. Ensuite nous donnons une explication de deux termes clés présents dans l'étude des artefacts, les *modules* et les *paramètres*. Enfin nous utilisons les travaux de [Malloch *et al.*, 2006], [Vinet, 2003], [Orlarey, 2009] et [Winograd *et al.*, 1994] pour éclairer une certaine structuration en niveaux d'abstraction, présente dans l'artefact.

1 Système Homme Artefact Son

Dans notre démarche visant de prime abord à étudier les artefacts, une première difficulté nous paraît être liée au premier point de singularité : l'artefact possède des apparences. Ce constat revient à dire que l'artefact ne peut se réduire à du code, mais qu'il concerne aussi, dans la situation d'analyse, ce que l'analyste hérite de la compréhension du code et, parallèlement, dans la situation compositionnelle, des idées, des intentions du compositeur.

Pour affronter cette difficulté nous proposons de considérer l'artefact comme un élément d'un système, qui en inclut deux autres, l'Homme et le Son. Nous amenons ci-après ce système à l'aide du principe de la boîte noire.

1.1 Boîte noire

1.1.1 Introduction

La boîte noire peut être comprise dans la situation qui s'installe entre l'analyste et l'artefact. Une première volonté qui consisterait à comprendre un artefact *in extenso*, à l'aide d'une lecture entière de son apparence, nous paraît vaine. En effet, comme nous l'avons vu, l'artefact résulte d'une activité de *design*, il contient d'innombrables idiosyncrasies, et possède une sémantique qui lui est propre. Avant de vouloir atteindre le fonctionnement global, ou même partiel, de l'artefact, il est donc nécessaire de se confronter à son apparence, d'intégrer le réseau de connexions sous-jacent et par la même la sémantique. Au lieu de partir de sa totalité, il paraît plus judicieux d'isoler un fragment, de le comprendre, puis de le rattacher à ce à quoi il est relié, et ainsi de suite.

Dans cette approche ascendante (*bottom up*), l'outil théorique de boîte noire nous semble particulièrement approprié. Théorisé au milieu du XXe dans le célèbre ouvrage *An introduction to cybernetics*, [Ashby, 1957], cet outil émerge d'abord au sein d'un contexte d'ingénierie :

The engineer is given a sealed box that has terminals for input, to which he may bring any voltage, shocks, or other disturbances he pleases, and terminals for output, from which he may observe what he can. [Ashby, 1957, p.86]

1.1.2 Éclairer le fonctionnement par le comportement

L'idée forte de cette théorie est d'appréhender cette boîte non pas en jetant de la lumière à l'intérieur, en regardant de quoi elle est faite – sa structure interne – mais en la regardant telle

quelle, fermée, et en comprenant ce qu'elle fait par l'observation de ses entrées et sorties, en les mettant en relation : « de relationner *inputs* et *outputs* sans entrer toutefois dans le mystère de la boîte noire » [Morin, 2005, p. 49].

Ainsi Ross Ashby marque-t-il une différence entre ce qu'*est* et ce que *fait* la boîte :

[Cybernetics] does not ask “what is this thing?” but “what does it do?” Thus it is very interested in such a statement as “this variable is undergoing a simple harmonic oscillation”, and is much less concerned with whether the variable is the position of a point on a wheel, or a potential in an electric circuit. [Ashby, 1957, p. 1].

Cette méconnaissance sur l'intérieur de la boîte, qui peut aussi être de la volonté de l'analyste¹⁵, se symbolise évidemment par la couleur noire de la boîte.

1.1.3 Au-delà de l'ingénierie

Une des forces de cet outil, et par là même de la cybernétique, se trouve dans le fait qu'il peut être appliqué indifféremment sur une boîte faite de composants électroniques, électriques, mécaniques ou encore psychiques :

Though the problem arose in purely electrical form, its range of application is far wider. The clinician studying a patient with brain damage and aphasia may be trying, by means of tests given and speech observed, to deduce something of the mechanisms that are involved. *Ibid.* p. 86.

Dès lors que l'intérieur de la boîte est ignoré, il devient possible d'appliquer l'outil indépendamment de « la nature » de la boîte, indépendamment de ses composants internes, et de même indépendamment de l'environnement des entrées et sorties.

1.2 Le système Homme Artefact Son (HAS)

1.2.1 Système cybernétique

Un autre intérêt de la boîte noire se trouve dans la mise en évidence d'un système. En reprenant la situation standard où un analyste (un agent) cherche à élucider le fonctionnement d'un fragment de code (une boîte noire), la cybernétique nous amène en effet à considérer, inévitablement, un système :

15 « A theory or an object becomes a black box when its ingredients cease to be visible or remarkable and turns into a tool that *does not need to be analysed anymore* » [Magnusson, 2009, p. 65] (nous soulignons).

By thus acting on the Box, and by allowing the bow to affect him and his recording apparatus, the experimenter is coupling himself to the Box, so that the two together form a system with feedback [Ashby, 1957, p. 87].

La boîte noire s'applique en effet quand il est possible d'agir sur les entrées de celle-ci, de sorte ensuite à pouvoir observer leurs conséquences sur les sorties. C'est donc par l'action sur les entrées et l'observation des sorties que des connaissances sur le fonctionnement de la boîte sont inférées par l'agent : entre l'agent et la boîte il y a donc interaction, tous deux forment un système avec une boucle de rétroaction (*feedback*).

1.2.2 Du sonore

Évidemment après avoir aperçu les bribes d'un tel système il nous faut prendre de la distance avec cette cybernétique. En effet avec cette dernière si les entrées et sorties de la boîte peuvent être de n'importe quelle « nature » elles doivent, en revanche, être discrétisées. L'éclairage de la boîte se fait alors avec une comparaison d'états discrets, par l'application de règles formelles sur des symboles. Or, dans notre contexte, la sortie de la boîte peut être uniquement sonore, continue¹⁶ et perceptive, mettant ainsi à mal l'application de règles formelles. Cela devient plus frappant lorsque l'on considère le système avec le compositeur : les ultimes sorties de la boîte ne sont alors autres que les sons, le matériau sonore de sa composition.

1.2.3 Homme, Artefact, Son

Nous voyons que l'analyse qui s'appuie sur l'artefact, qui consiste au moins dans un premier temps à le comprendre, se réalise au sein d'un système fait de trois éléments : l'Homme, l'Artefact et le Son¹⁷.

Ce système se justifie dans notre démarche par deux interactions. Il y a d'une part interaction entre l'Homme et l'Artefact. Le compositeur est un programmeur, c'est lui qui écrit, modifie, assemble le code, et l'analyste se doit dans un premier temps de comprendre cette programmation effectuée. Et d'autre part, inévitablement, il y a interaction entre l'Homme et le Son. Le compositeur façonne son matériau, le perçoit, et l'analyste l'intègre au cours de son travail.

Enfin, l'Artefact et le Son sont liés par une relation plus simple, une relation technologique. Avec une compréhension téléologique on peut comprendre l'Artefact comme un tout qui a

16 L'état discret des entrées/sorties de la boîte noire – une suite de quelques *symboles* – n'est pas du même ordre de grandeur que l'état discret d'un signal audionumérique – une suite de dizaine de milliers d'échantillons – qui coexiste au temps (cf.[Risset, 1996] par exemple). C'est pour cette raison que nous ne considérons pas le son « qui sort de la boîte » comme discret, bien qu'il le soit technologiquement. Nous revenons ultérieurement sur ce point dans le titre 3.1.2 de ce même chapitre.

17 Pour démarquer les éléments de ce système nous les marquons par la suite d'une majuscule.

pour fin de produire le signal audionumérique de la pièce, qui sera ensuite diffusé par l'intermédiaire de deux transductions classiques¹⁸. La production de ce signal se fait, à un niveau infinitésimal, par le *calcul* d'échantillons audio, par la transformation d'un code, plus ou moins textuel¹⁹, en une suite d'échantillons qui, dans le temps, vont former le signal audionumérique. Afin d'appuyer cette présence de calcul dans la production du son, nous qualifions cette relation de *computationnelle*.

Ainsi pouvons-nous représenter le système Homme Artefact Son (HAS).

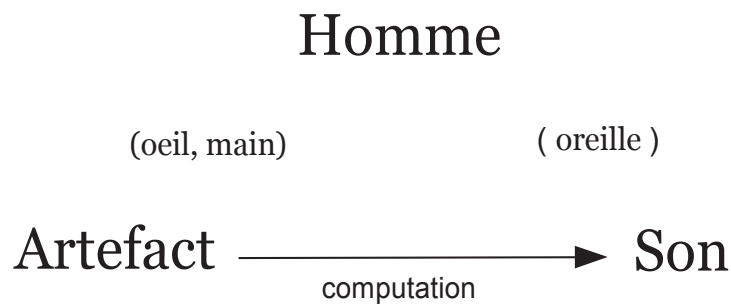


Fig.3 - Le système HAS

18 Signal audionumérique vers signal analogique (CNA : Convertisseur Numérique Analogique), puis signal analogique vers onde sonore (Haut-Parleurs).

19 L'opposition « graphique » contre « textuel » pour qualifier les langages de programmation s'estompe quelque peu puisque du texte est inévitablement présent dans les langages « graphiques » (e.g. le nom des boîtes dans Max).

2 Éléments de l'Artefact

Nous mettons ici au clair trois éléments essentiels dans l'étude des artefacts, les *signaux audio*, les *modules* et les *paramètres*. Nous pouvons situer la succession de ces éléments au sein du système HAS : le premier est limitrophe au Son, le deuxième proche de l'Artefact, et le dernier davantage proche de l'Homme.

2.1 Signal audionumérique

Les signaux audio²⁰ sont évidemment très récurrents dans l'étude des artefacts. Une de leurs particularités est qu'ils peuvent avoir un sens différent selon les types de métiers, entre compositeur et programmeur par exemple. Ci-après nous présentons d'abord une explication technologique puis nous en apportons une plus singulière, qui prend sens avec la perception.

2.1.1 Vue technologique

Le sens premier d'un signal audio nous semble être technologique. Il découle de la théorie de l'échantillonnage, amenée par Claude Shannon en 1949, qui rend théoriquement possible la transformation d'un flux de nombre (un signal discret) en une forme d'onde analogique :

A sound wave with frequencies from 0 to B cycles per second can be generated from a sequence of two B pulses per second. [Mathews, 1963, p. 553].

Un signal audionumérique est ainsi un signal de données qui possède une fréquence de définition (fréquence d'échantillonnage), fixe, de sorte à permettre sa transformation en une onde sonore. À cette condition il faut ajouter que les données du signal doivent être quantifiées avec une définition suffisante.

La particularité d'un tel signal, qui respecte ces conditions, est qu'il peut tout à fait ne pas être sonore. On peut par exemple penser à une sinusoïde dans l'infra audio, ou encore une composante continue, sans oscillation, de valeur fixe (DC offset). Dans de tels cas, bien que les conditions technologiques soient respectées, le signal audio n'a rien de sonore.

Cette définition technologique est celle présente dans les langages de programmation (e.g. Pure Data). Comme nous le verrons dans ce chapitre, ces signaux audio s'opposent à ceux de données, typiquement d'un débit plus faible.

²⁰ Nous utiliserons cette forme abrégée pour ne pas alourdir.

2.1.2 Un signal audible

Pour faire face à cette possibilité paradoxale où un signal audio peut ne pas être sonore, nous apportons notre propre définition :

Un signal audio est un flux de données qui, s'il était adressé à un haut-parleur, engendrerait un son perceptible.

Un tel signal est donc un son en puissance. C'est un signal qui, à l'aide d'une simple manipulation (deux transductions en séries) peut devenir sonore. Tout l'intérêt de cette définition se trouve dans le fait que quelque chose de perçu, un son, puisse être *représenté* par quelque chose d'autre, un signal. Notre définition hérite ainsi des travaux d'étude sur les technologies de production du son qui ont fait le XXe (e.g. [Genevois *et al.*, 1999]). Dans ces technologies, l'intérêt peut-être premier est, selon nous, qu'elles ont pu transformer le son en quelque chose d'autre, quelque chose qui ne soit plus un phénomène dissipatif, évanescent, coexistant au « temps présent », et inévitablement inscrit dans un espace-temps. Avec ces technologies – et les transductions qu'elle sous tendent – le son est devenu quelque chose que l'on peut voir et sur lequel on peut agir. C'est à ces conditions de visibilité et d'action, que nous parlons de « rendre présent », de représentation²¹ : un signal audio représente un son.

Dès lors, dans la suite de notre travail, quand nous parlons de signal audio c'est dans ce dernier sens. À l'inverse, pour se référer au sens technologique, nous emploierons l'expression « signal audio technologique ».

2.2 Module et paramètre

Les termes *module* et *paramètre* sont aussi quasi inévitables dans l'étude des artefacts. Nous les expliquons une première fois avec le sens qu'ils ont hérité au sein du langage Music N, essentiellement à partir de l'ouvrage pionnier de Max Mathews, *The technology of computer music*, [Mathews, 1969], puis une seconde fois, plus généralement, en tant qu'outil pour investir les artefacts.

2.2.1 Le langage historique Music N

Durant la seconde moitié du XXe le langage Music I a évolué pour donner Music II, Music III, etc., de sorte à former une famille de langages. Le terme Music N est ainsi utilisé pour se référer à cette famille.

²¹ cf. [Duchez, 1980].

2.2.1.a Un problème

Cette nouvelle représentation du son, autorisée par l'application de la théorie de l'information, est au cœur des problèmes qu'affronte le premier langage de programmation musicale. En effet ce langage est présenté par son concepteur Max Mathews (cf. [Mathews, 1963], [Mathews, 1969]) comme la réponse à un problème : l'Homme musicien ne peut agir directement sur cette nouvelle représentation du son.

To specify individually 10 000 to 30 000 numbers for each second of music is inconceivable. [Mathews, 1963]²²

Dès lors, le travail du concepteur peut être compris comme l'invention d'un moyen pour permettre à l'Homme d'engendrer, de produire, ce signal audionumérique. Et dans la réalisation de ce moyen, le concepteur doit faire en sorte de ne pas limiter la production de son possible : l'enjeu est de pouvoir « spécifier n'importe quelle séquence de son »²³.

Enfin, dans cette réalisation, si la spécification de tous les nombres nécessités par un signal est impensable, elle reste, néanmoins, un idéal :

The most powerful and universal possibility would be to write each of the millions of samples of the pressure wave directly. This is unthinkable. [Mathews, 1969, p.35]

2.2.1.b Une solution

La solution amenée par M. Mathews consiste en l'articulation de deux éléments clés : des « unit-generators », « building blocks for sound-synthesizing instruments to provide great flexibility », et le concept de « note », « for describing sound sequences » [Mathews, 1969, p.34].

La distinction de ces deux derniers se retrouve dans l'organisation même du langage. Les « unit-generators » s'interconnectent de sorte à former des *instruments*, et ceux-ci concernent une partie du langage nommé orchestre (*orchestra*). Les notes se trouvent elle dans l'autre partie du langage, nommée, pour poursuivre l'analogie, partition (*score*).

22 Pour compléter ce problème de scalabilité on peut ajouter que ces nombres devraient être définis parmi un ensemble de 256 valeurs – en quantification en 8 bits.

23 « He [the composer] would like to have a very powerful and flexible language in which he can specify any sequence of sounds. » [Mathews, 1969, p. 35].

Unit-generator

Les *unit-generators*²⁴ (ou UGen) sont des unités qui génèrent ou effectuent des opérations sur des signaux :

The unit generators perform functions that experience has shown to be useful. For example, there are oscillators, adders, noise generators, and attack generators. Many unit generators perform conceptually similar functions to standard electronic equipment used for electronic sound synthesis (*Ibid.* p.36)

C'est en combinant ces Ugen, en les interconnectant, que le compositeur va créer ses propres producteurs de signaux audio, soit, dans la terminologie Music N, ses propres *instruments*.

Note

Les *notes* sont des événements qui s'articulent avec les Ugens au travers d'une suite de *paramètres*. Ces derniers découlent de l'élaboration d'un instrument par le compositeur. Ils lui permettent de contrôler la production sonore, d'avoir la main sur le phénomène sonore.

In addition to audio input and/or output, a Ugen may support a number of control inputs that control parameters associated with the Ugen. [Wang, 2007, p. 12].

En plus d'intégrer ces paramètres, les notes incluent trois autres données imposées par le langage : le numéro de l'instrument auquel la note se destine, le temps de départ et la durée de la note²⁵.

2.3 Au-delà de la technologie

2.3.1 L'influence de Music N

L'articulation entre Ugen, paramètres et notes, qui a permis les premières synthèses sonores, dépasse en fait largement la famille Music N. Comme cette famille a su s'adapter aux nombreuses évolutions informatiques de la seconde moitié du XXe, les idées de conception qui ont présidé à la définition du langage, ont aussi influencé les nouveaux langages de

24 Pour être précis, les Ugen sont arrivés avec Music III et non pas avec Music I (cf. [Wang, 2007, p. 12]).

25 Ces trois données participent à ce que nous avons appelé, au chapitre précédent, les *conventions* d'un langage de programmation.

programmation musicale comme en témoigne Ge Wang, qui a conçu le langage nommé ChucK :

[Music N] also introduced several key concepts and ideas which still directly influence languages and systems today [Wang, 2007, p. 11]

Parmi les grandes idées de Music N nous pouvons relever celle de la modularité, l'interconnexion entre Ugens, comme le relève J.-C. Risset :

En fait, la conception modulaire de Mathews a marqué la plupart des programmes de synthèse – comme Music 360, Music 11, CMusic, Csound – et des synthétiseurs analogiques ou numériques – comme ARP, DX7, 4A, 4B, 4C, 4X, Syter – aussi bien que des langages de simulation de circuits électroniques, et plus tard un langage de créations d'interactions temps réel comme Max. [Risset, 1995, p. 23].

2.3.2 Des Ugens aux modules

Il nous semble qu'un certain glissement s'est opéré dans le champ de l'informatique musicale concernant le terme *Ugen*. Ce glissement, que nous ne faisons ici qu'effleurer, consiste en la substitution du terme *module* à celui de *unit-generators*. Elle est présente par exemple chez Julius Smith :

A unit generator is a fundamental building block, or module, used to build a variety of sound-generating algorithms. [Smith, 1991]

Ou encore chez J.-C. Risset dans un article moins spécialisé :

L'utilisateur est laissé libre de décider du type de synthèse sonore qu'il veut mettre en œuvre : il choisit des modules dont chacun correspond à une production ou une transformation sonore [...], et il les assemble à loisir, comme s'il "patchait" un synthétiseur modulaire. [Risset, 1999]

Une piste pour éclairer cette substitution est peut-être de relever que le terme *unit-generator* ne rend pas compte de la diversité de ces unités. En effet, comme l'illustre la dernière citation, en plus de pouvoir générer, ces « unités » peuvent aussi transformer un signal déjà existant.

2.3.3 Pour investir l'artefact

Au-delà de la solution apportée par Music N, nous observons que l'articulation *module* et *paramètre* est une aide pour investir l'artefact. L'intérêt de cette articulation, pour nous, est que ces deux termes ne sont pas propres à un langage, ni à une pratique, mais qu'ils s'avèrent un moyen pour comprendre globalement l'outil que le compositeur s'est façonné.

Il faut noter que *module* ne fait pas strictement référence à la pratique de la composition sur synthétiseur modulaire, trop circonscrite à nos yeux en regard de l'activité de *design* qui est – ou peut être du moins – impliquée avec les langages de programmation²⁶.

Le terme *module* est ainsi utile au-delà de considérations technologiques, et au-delà de l'implémentation d'un langage. Il peut être utilisé pour comprendre la production sonore/musicale réalisée par le compositeur. Nous retrouvons par exemple cet usage dans [Schnell *et al.*, 2002], où le concept « d'instrument composé », qui permet une saisie des « instruments de musique numériques », est défini comme un assemblage de modules :

A composed instruments can be seen as composed of streams running through processing modules of these four [analyse, synthèse, traitement et mémoire] categories. [Schnell *et al.*, 2002]

Dans la suite de nos travaux ces articulations entre modules et paramètres s'avéreront essentielles, si bien que nous les définissons au chapitre IV. Nous laissons donc pour l'instant volontairement flou le type de fonctions que les modules peuvent effectuer.

²⁶ On peut en effet constater que les modules des synthétiseurs modulaires (oscillateurs, filtre etc.) s'avèrent donnés par le fabricant, quand ils peuvent être construits par le compositeur avec les langages.

3 Des dimensions

Le couple de mots paramètres/modules, qui peut être signifiant aussi bien dans le référentiel d'un langage de programmation que dans des écrits théoriques visant à étudier les artefacts, nous laisse penser que l'artefact n'est pas « quelque chose de plat », quelque chose qui ne posséderait qu'une seule voie pour être investi. A *contrario* on s'aperçoit qu'il est fait d'une importante densité qui peut s'éclairer à l'aide de dimensions. Nous proposons ci-après d'éclairer l'artefact à l'aide de quatre dimensions, regroupées deux à deux. Le premier groupe concerne des dimensions technologiques qui sont en lien avec le *calcul* de l'ordinateur, le second des dimensions cognitives qui prennent sens avec les interactions de l'Homme.

3.1 Dimensions technologiques

3.1.1 Donnée ~ audio²⁷

Comme nous l'avons vu précédemment il est d'usage de différencier les signaux audio de ceux de données. En fait, dans les implémentations des artefacts ces signaux audio technologiques sont les seuls éléments à être tributaires de lourdes contraintes technologiques. C'est parce qu'ils doivent passer l'étape de la transduction numérique vers analogique, qui obéit elle-même aux lois de l'échantillonnage, qu'ils sont tributaires de ces règles strictes (fréquence échantillonnage et quantification pour l'essentiel). À la différence, de telles restrictions sont absentes des autres éléments de l'artefact. Ainsi peut-on définir les signaux de données par la négative, ce sont des signaux qui ne sont pas définis à la fréquence d'échantillonnage.

Cette distinction se retrouve dans la conception même de nombreux langages de programmation. Ainsi de Csound et Pure Data par exemple :

- Dans Csound cette différence est présente dans le *typage* de variable. Les variables dédiées aux signaux audio sont nommées *a-name*, tandis que celles dédiées aux données sont nommées *k-name*. Chacune a des fonctionnements fort différents :

i- and k-rate variables are scalars (i.e. they take on only one value at any given time) and are primarily used to store and recall controlling data (...). a-rate variables, on the other hand, are arrays or vectors of information. (...) [they] represent a finer resolution of time. (Manuel Csound v6 : Types, Constants and Variables Syntax of the Orchestra²⁸).

²⁷ Nous empruntons à [Mounin, 1997] le caractère tilde (~) pour marquer, non pas l'opposition, mais la richesse sémantique qui se dessine entre deux mots.

²⁸ <http://www.csounds.com/manual/html/OrchKvar.html>, visité le 10 07 2017

- Dans Pure Data la distinction est aussi présente. Toutes les boîtes (ou fonctions) qui traitent/génèrent des signaux audio (en entrées ou en sorties) finissent par le caractère tilde (e.g. [osc~] pour la production d'une sinusoïde), et les signaux audio sont représentés par des câbles d'une plus grande épaisseur que ceux de données. Dans la documentation du langage, [Puckette, 2005], le concepteur explique simplement que les signaux de données sont sporadiques, à la différence des signaux audio qui eux sont des suites continues de nombres²⁹.

Enfin, il nous paraît important de préciser que cette distinction est *technologique*. Elle est en effet présente à des fins de minimiser la charge processeur :

The chief reason for this separation [audio vs. control rate] is computational efficiency [Wang, 2007, p. 19].

Cette distinction ne nous paraît pas comme une solution apportée par Max Mathews face au problème de la conception d'un langage de programmation musical, mais comme une solution face à un problème technologique.

3.1.2 Discret ~ continu

La dimension donnée ~ audio, parfois présente dans la conception même des langages, en appelle une autre, plus grande. L'adjectif *sporadique* présent dans la documentation de Pure Data nous amène en effet à différencier des événements discrétisés, non cycliques, d'événements cycliques, comme les signaux audio qui obéissent à une fréquence d'échantillonnage. Similairement, on retrouve cette distinction entre un phénomène discret et un continu dans la famille Music N, *via* le concept de *note* :

Sound exists as a continuous function of time starting at the beginning of a piece and extending to the end. We have chosen, for practical reasons, to chop this continuous sound into discrete pieces, called notes, each of which has a starting time and a duration time. [Mathews, 1969, p. 36]

Il est important de relever que cette distinction n'est pas significative au même niveau d'implémentation que ne l'est la précédente, entre donnée et audio. La dimension discret ~

29 « Audio signals aren't sporadic; they are continuous streams of numbers » [Puckette, 2005]. La continuité des signaux audio se fait par le calcul d'une suite de bloc d'échantillons, nativement 64. « When audio computation is turned on (...) Pd sorts all the tilde objects into a linear order for running; then this linear list is run down in blocks of 64 samples each; at 44100 Hz. this means the audio network runs every 1.45 milliseconds. ». *Ibid.*

continu est davantage « floue », elle n'est en effet pas assez précise pour rendre compte de la gestion des données au sein du système. De plus, on peut noter qu'au sein de cet environnement numérique tout est nombre, tout est discret : un signal audio n'est (technologiquement) qu'une suite de nombres par unité de temps. Il faut alors voir la présente dimension discret ~ continu davantage proche de la théorie, comme une dimension utile pour l'analyste qui investit l'artefact.

3.2 Dimensions cognitives

La précédente dimension nous invite à sortir de considérations technologiques, centrées sur le « bas du système », pour se diriger vers le haut, qui concerne l'Homme et sa cognition. Dans cette direction nous témoignons de deux autres dimensions. La première est extraite d'un article sur les « instruments de musique numérique » (Digital Music Instrument, DMI par la suite), [Malloch *et al.*, 2006], qui présente trois domaines d'interaction entre le musicien et les DMI : le domaine du *signal*, du *signe* et du *symbole*. La seconde dimension est plus répandue, il s'agit de différencier le *symbole* du *signal*. Nous la présentons en reprenant l'article [Vinet, 2003] avec l'idée de « connaissance de contenu » (*content aware*).

3.2.1 Symbole, signe et signal

Les travaux de [Malloch *et al.*, 2006] visent à établir un cadre conceptuel (« conceptual framework ») pour l'analyse et la conception des DMIs. Les auteurs présentent trois types d'interactions qui ont lieu entre l'Homme (le performeur) et l'Artefact (le DMI). Ces distinctions sont amenées à partir de travaux réalisés par le chercheur Jens Rasmussen portant sur les interactions homme-machine³⁰. Trois comportements d'interaction sont repris : *skill-based*, *ruled-based* et *model-based* :

Briefly, *skill-based* behavior is defined as a real-time, continuous response to a continuous signal, whereas *ruled-based* behavior consists of the selection and execution of stored procedures in response to cues extracted from the system. *Model-based* behavior refers to a level yet more abstract, in which performance is directed towards a conceptual goal and active reasoning must be used before an appropriate action -rule- or skill- based) is taken. [Malloch *et al.*, 2006, p. 1].

À chacun de ces modes d'interaction les auteurs associent trois catégories « d'information traitées par l'Homme » (human information processing). Respectivement les *signes*, les

³⁰ J. Rasmussen, *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*, Elsevier Science Inc., New York, USA, 1986

signaux et les *symboles*. Enfin, des associations entre ces modes d'interaction et la performance musicale sont réalisées :

- Le domaine du *signal* est celui, inévitablement, du continu. Il concerne le contrôle de phénomènes perceptibles dans le temps de la performance. Il s'illustre bien par exemple avec un des premiers instruments électroniques, le Thérémine, pour lequel une correspondance est présente entre les gestes de la main de l'instrumentiste et la perception sonore (typiquement la dimension horizontale correspond à la dimension des hauteurs).
- Le domaine du *signe* concerne la mise en ordre, le jeu de séquences musicales *prédéfinies*. Il s'agit ainsi de contrôler des processus plutôt que des signaux perceptifs, ces processus ayant été définis en amont du moment de la performance.
- Le domaine du *modèle* est celui où le performeur, en termes de quantité, a le moins de contrôle. Il s'agit de l'activité la plus forte cognitivement dans le sens où le performeur a un but à atteindre et a des représentations sur les moyens qu'il doit mettre en œuvre pour atteindre ce but (« internal representations of the task » ; « problem solving and planning are required »). Les auteurs prennent l'exemple des partitions conceptuelles où, à défaut de notes sur portées, les performeurs doivent rendre musicales des indications textuelles.

Ces types d'interactions s'avèrent aussi éclairants en dehors d'une étude des DMI stricte. Nous pouvons par exemple les illustrer avec la pièce *Jupiter* que nous analysons dans la dernière partie de cette thèse. Dans la section VI, où essentiellement l'électronique est faite d'un déplacement d'une enveloppe spectrale sur une sommation compacte de sinusoïdes, on retrouve une interaction du domaine du *signal*. Ensuite, les importantes interpolations réalisées par un sampler (section V par exemple), très singulières à la pièce, appartiennent au domaine du *signe*³¹. Enfin, le domaine du *symbole* nous paraît lui plus difficile à illustrer. Il est concerné par l'activité de composition. Dans *Jupiter* il pourrait correspondre au moment où le compositeur rend effectif et développe cette idée d'enveloppe spectrale interactive. Dans *Inharmonique* de J.-C. Risset ce domaine correspondrait au travail d'implémentation des structures de cloches, lorsque le compositeur a eu une « représentation interne » de la réalisation de l'idée.

31 Le compositeur emploie d'ailleurs le terme *neume* pour expliquer un aspect de ces interpolations [Manoury, 2017].

3.2.2 Symbole et signal

Les travaux de [Malloch *et al.*, 2006] peuvent être vus comme l'affinement d'une autre distinction, plus répandue, qui consiste à différencier les *symboles* des *signaux*. Cette distinction est évidemment aussi en rapport avec la précédente dimension discret ~ continu mais, comme nous le montrons ici, elle a la particularité de s'inscrire avec des considérations essentiellement musicales. Nous nous basons ci-après sur un article de Hugues Vinet, *The representation levels of music information*, [Vinet, 2003].

Avec l'auteur nous pouvons d'abord comprendre cette distinction comme le fruit d'une volonté de saisir les différentes représentations présentes dans les outils de création – à l'aune des années 2000. Dans ces outils on se rend aisément compte que deux types de représentations priment : d'une part celles qui simulent l'environnement traditionnel hérité de la théorie musicale, illustré par la note sur sa portée ; et de l'autre celles qui rendent présents le son par sa forme d'onde. Les premières sont qualifiées de *symboliques* et les secondes de représentations de *signaux*.

Content aware

Pour distinguer ces deux représentations maîtresses, en plus de la distinction évidente discret ~ continu, l'auteur a recours à une notion de « connaissance de contenu » (*content aware*) :

the symbolic representation is *content-aware* and describes events in relation to formalized concepts of music (music theory), whereas the signal representation is a blind, content-unaware representation [*Ibid.*, p. 194]

En ayant en tête des simples notes disposées sur une portée nous pouvons comprendre le terme « connaissance de contenu ». Selon nous ces représentations sont qualifiées ainsi puisqu'elles se basent sur un vaste ensemble de connaissances issu de la théorie musicale. Elles prennent sens avec ce vaste contexte (*background*). La représentation qui est engendrée, le motif que dessinent ces quelques notes, est qualifiée de « connaissance de contenu » dans la mesure où elle s'appose sur ce contexte de la théorie musicale, qu'elle « connaît » donc. À la différence, les représentations de *signaux* correspondent à ce que nous avons appelé des *représentations technologiques* : la représentation d'une forme d'onde peut être faite sur n'importe quel signal, elle est bâtie non pas sur des connaissances, qu'elles ignorent (*content unaware*) donc, mais sur des échantillons, qui résultent de l'application technologique du théorème de l'échantillonnage.

Les notes de la partition traditionnelle sont un bon moyen pour illustrer ce que l'on entend par *représentations symboliques*. Néanmoins, il ne faut pas réduire ce type de représentation

aux seules notes. Les représentations symboliques peuvent en effet déborder de la théorie musicale occidentale et être propre à une seule pièce d'un compositeur. C'est par exemple le cas pour la pièce *Solitudes* (2007), pour électronique, du chercheur compositeur Hans-Christoph Steinner. Cette pièce a été réalisée avec une possibilité propre au langage Pure Data, appelée « données structurées » (*structured data*, cf. [Puckette, 2002b] et [Dahan, 2005b]), ce qui a permis au compositeur de concevoir ses propres *représentations* nécessaires à la composition de sa pièce.

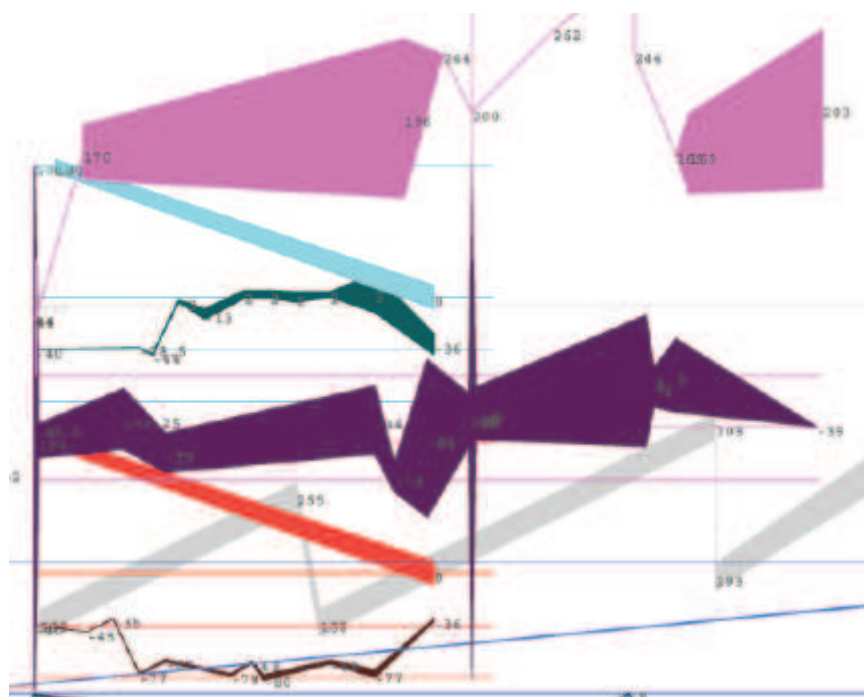


Fig. 4: Exemple de représentation pour la pièce *Solitude* (2007) de Hans-Christoph Steinner

Cette figure est un extrait des représentations des structures de données conçues pour composer la pièce. Le compositeur explique les significations de ces représentations (e.g. les couleurs correspondent à des samples et la dimension verticale est utilisée une fois pour contrôler leur playback) sur sa page web³². L'importance accordée aux représentations graphiques dans ce travail nous rappellent les travaux de Xenakis avec le système UPIC (cf. [Marino, 1993]), mais les questions sont ici réinvesties, 30 ans après, avec le *design* que permet un langage de programmation.

32 « In the score, time flows from left to right. Each color represents a sample. Each sample controller has two arrays: the brighter, bigger one on top controls sample playback; and a smaller darker one at the bottom controls amp and pan. The lowest point of the sample array is the beginning of the sample, the highest is the end, and the height of the array is how much and what part of the sample to play starting at that point in time. There are between 50 and 100 voice polyphony for the samples. The height of the amp/pan array is the amp, and the y location is the pan. » (<https://at.or.at/hans/solitude/>, visité le 10 07 2017).

4 Des niveaux d'abstraction

Le concept de multi niveaux dans l'explication scientifique est très récent et représente une des principales innovations des sciences cognitives.

F. Varela, 1988

Une abstraction est le résultat d'une opération par laquelle on conçoit quelque chose séparément d'un ensemble d'autres choses formant un tout et qui, dans la réalité ou dans les représentations habituelles, ne peuvent être séparées de cette chose.

F. Varenne, 2009

Les précédents types de représentation, symbole et signal, ne sont pas sans lien avec deux autres niveaux chers aux sciences cognitives : le niveau symbolique et sub-symbolique. Dans le contexte d'une pensée cognitiviste (~1970) il a en effet été d'usage d'appréhender la cognition de l'Homme à l'aide de ces deux niveaux (cf. [Varela, 1988]). Ces rapprochements entre *symbole* et *signal* en musique, et *symbolique sub-symbolique* en sciences cognitives, sont par exemple présents dans [Leman, 1993], et repris dans [Bresson, 2007]. Chez ce dernier notamment le niveau symbolique est signifiant pour le compositeur, et les données concernées sont « susceptibles d'être mises en œuvre dans des opérations musicales ». À la différence, le niveau du sub-symbolique rassemble les « processus et représentations dont la signification relève du traitement informatique », et il permet « d'accéder à la réalité sonore » (*ibid.* p. 90).

Cette vue en différents niveaux se fait donc au sein d'un champ de forces conséquent, où s'exercent, au moins, la cognition et la perception de l'Homme, les représentations de la théorie musicale occidentale, et les niveaux de computation en informatique. Nous investissons ci-après ce champ en trois mouvements. D'abord nous utilisons une dernière fois la différence symbole ~ signal pour apporter de la lumière sur le principe d'abstraction et de niveaux, ensuite nous effectuons une excursion dans les entrailles de l'ordinateur pour présenter, à partir de [Winograd *et al.*, 1994], cinq niveaux d'abstraction généraux. Enfin nous présentons une « vue informationnelle » qui permet de traverser ces différents niveaux.

4.1 Niveaux d'abstraction et connaissance

Nous pensons que la notion d'abstraction, quoique absconse de prime abord, peut expliquer clairement cette nuance entre des représentations symboliques et de signaux que nous avons vue précédemment.

En reprenant l'exemple de [Vinet, 2003], une représentation est dite symbolique lorsqu'elle est bâtie sur la théorie occidentale. Elle fait sens pour « l'Homme musicien » qui a intégré, et maîtrisé, les concepts de cette théorie (*e.g.* la note, l'accord, la gamme, la clé, la signature, la notation rythmique). Ce qui est essentiel c'est que ces représentations sont signifiantes à la condition que *quelque chose d'autre* soit acquis. Du simple fait que quelque chose d'autre soit nécessaire pour que la compréhension ait lieu, nous qualifions ces représentations d'un plus *haut niveau d'abstraction* que d'autres, qui ne nécessiteraient pas « quelque chose d'autre » pour être signifiantes (*e.g.* les représentations de signaux).

Nous pouvons illustrer ce principe d'abstraction de façon concrète à l'aide de l'exemple de structure de cloche d'*Inharmonique*, que nous avons présenté dans le précédent chapitre. Le numéro de structure de cloche – que J.-C. Risset doit spécifier lors de la composition – est d'un niveau d'abstraction plus élevé que, par exemple, le paramètre de fréquence d'un oscillateur. En effet, pour que le numéro de structure de cloche soit signifiant, il faut avoir en tête la répartition des partiels de la cloche à simuler, soit sa sonorité. À la différence, la fréquence d'un oscillateur ne nécessite que très peu de connaissances, seulement les fondamentaux de l'ingénierie audio : c'est un paramètre de bas niveau. Ou mieux, le numéro de structure de cloche est d'un plus haut niveau d'abstraction puisque pour être signifiant il est nécessaire que la signification de la fréquence d'un oscillateur soit acquise.

Dans ce contexte, les niveaux d'abstraction sont ainsi relatifs à la quantité de connaissances, ou de significations, nécessaire pour que des représentations soient compréhensibles.

Nous comprenons ainsi la différence entre des représentations symboliques et de signaux avec le principe même d'abstraction. Les représentations symboliques sont d'un niveau d'abstraction plus élevé que celles de signaux, nous pouvons les voir comme bâties sur des niveaux moins élevés, dont le signal.

4.2 Niveaux d'abstraction en informatique

Les niveaux d'abstraction sont ainsi fortement présents en informatique. Ils sont certainement plus habituels pour qui est familier avec des langages de programmation³³. Dans

³³ Comme nous l'avons vu, dans Pure Data le terme même *abstraction* est le nom d'un procédé inhérent au langage, telle une réification.

un premier temps nous exposons brièvement cinq niveaux d'abstraction présents dans l'usage de l'informatique, identifiés et définis dans le célèbre ouvrage *Understanding computers and cognition : A new foundation for design*, [Winograd et al, 1994, p. 87]. Puis, dans un second, nous effectuons des parallèles entre ces niveaux généraux et le langage Music N.

4.2.1 Du transistor au schème

- La machine physique

Ce premier niveau est fait des composants de la machine (carte mère, disques, processeurs ...) qui eux-mêmes se décomposent en un ensemble de sous-composants. La particularité de ce niveau est que cette décomposition est récursive : les sous-composants se décomposent en structures de composants et ainsi de suite, jusqu'à obtenir « the basic physical elements, such as stands of copper and areas of semiconductor metal laid down on a wafer of silicon crystal » *Ibid.* p. 87

- La machine logique

Ce niveau correspond aux portes logiques bâties sur les précédents composants physiques. « certain range of voltages are interpreted as representing a logical 'true' and other ranges a logical 'false' ». À un premier sous-niveau, il s'agit des portes logiques et inverseurs, chers à l'automatique, puis à un second des multiplexeurs, des unités logiques d'arithmétique, des décodeurs d'adresse.

- La machine abstraite

Ce troisième niveau concerne les instructions machines : « simple operation of fetching or storing a symbol or performing a logical or arithmetic operation, such as a comparison, an addition, or a multiplication ». « it is the lowest level at which the programmer has control over the details of activity ».

- Le langage de haut niveau

Cet avant-dernier niveau concerne les langages informatiques généraux comme, à l'époque de la publication, FORTRAN, BASIC, COBOL et LISP, « which provide elementary operation at a level more suitable for representing real world domains ».

- Un schème de représentations pour les faits, « A representation scheme for 'facts' »

Ce dernier niveau est expliqué par les auteurs à l'aide d'un exemple issu de l'intelligence artificielle que nous reprenons.

Soit l'affirmation « Eric vit à Chicago » à stocker informatiquement. Si nous étions dans le niveau précédent, langage de haut niveau, le programmeur devrait coder cette affirmation avec une série de manipulations dans une base de données. À la différence, dans ce dernier niveau d'abstraction, plus haut, le programmeur ajouterait une *proposition* à une collection d'axiomes, comme « habite à », et ceci de sorte que cela ne soit pas réciproque entre les deux éléments (Chicago n'habite pas à Eric). À ce niveau il ne s'agit donc plus de variable booléenne ou de fonction conditionnelle, mais de *propositions*, élaborées par les programmeurs sur le précédent niveau. À partir de toutes les propositions stockées il serait ensuite possible, avec des règles d'inférences, d'en produire de nouvelles, comme « quels sont les voisins d'Eric » ?

Dans ce dernier niveau l'essentiel nous semble de comprendre qu'il est bâti sur le précédent afin d'atteindre des représentations qui appartiennent à notre langage, ainsi de la relation « habite à » dans le précédent exemple.

[this last level] use the symbol structures of a higher level language to represent facts about the world. *Ibid.* p. 89 (souligné dans le texte).

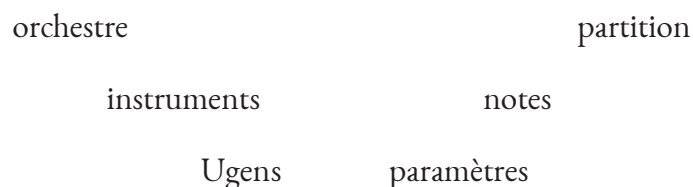
4.2.2 Music N, schème de représentations

Dans notre contexte nous devons considérer ces niveaux à partir des *langages de haut niveau*. Car évidemment, le compositeur comme l'analyste ne sont pas confrontés aux niveaux *infra*. Ceci peut s'illustrer notamment avec Music N. La maîtrise de Music III a pu se faire sans connaître les instructions de l'IBM 7090, et de même, la maîtrise de Music V a pu se faire sans connaissance du langage utilisé pour son implémentation, FORTRAN³⁴. Ainsi l'ouvrage historique présentant Music V, [Mathews, 1969], qui a pour fin, disons, d'apprendre à faire de la musique avec ce nouveau langage, n'intègre pas une explication des commandes réalisées en FORTRAN, mais bien des explications sur ce que peut faire le langage Music V, avec son propre référentiel, ses propres représentations. C'est bien à cette condition que l'on peut associer Music V au dernier niveau, à un *schème de représentations*.

The language implementation is opaque in that the detailed structure of computer systems that actually carry out the task are not relevant in the domain behavior considered by the programmer. [Winograd *et al*, 1994, p.86].

34 De Music I (1957) à Music IV (1963) inclus, l'implémentation s'est faite à partir du niveau *machine abstraite*, qui dépendait de l'architecture des énormes ordinateurs de l'époque (cf. [WANG 2007, p. 14]). Puis, à partir de Music V, l'implémentation s'est faite à partir du niveau supérieur, *langage haut niveau*, avec FORTRAN, ce qui lui permit de ne plus être tributaire d'une lourde architecture matérielle, soit de propulser sa diffusion.

Enfin, relevons que le schème de représentations de Music N se dessine au travers de la complémentarité de deux groupes de mots, que l'on peut ainsi représenter :



4.2.3 Une surface dans les niveaux informatiques

Cette rapide excursion dans les langages nous révèle qu'utiliser l'outil informatique c'est avoir recours, sciemment ou non, à différents niveaux d'abstraction. En plus des niveaux présents « dans la machine », il faut ajouter la présence de niveaux « au-dessus » du langage utilisé, de niveaux propres au compositeur ou designer, réalisés durant le travail. La différence entre les niveaux dessous et au-dessus se trouvent en effet dans leurs spécialisations. Si un langage de programmation est réalisé « à propos de quelque chose », pour un domaine d'application³⁵, un artefact musical, support à la composition, est lui réalisé pour une pièce musicale.

La classique métaphore de l'iceberg peut ici nous être utile. Les niveaux propres au compositeur, réalisés dans un ou quelques langage(s) spécialisé(s) (schèmes de représentations) seraient présents à la surface, et en dessous, immergés, on trouverait les niveaux moins spécialisés : des *langages de hauts niveaux* à la *machine abstraite*.

One of the properties unique to the digital computer is the possibility of constructing systems that cascade levels of representation one on top of another to great depth *Ibid.* p. 87

Les niveaux émergés sont donc véritablement le fruit du travail compositionnel.

Parmi les exemples que nous avons donnés dans le précédent chapitre il s'agit chez M. Puckette de l'abstraction Pure Data qui lui permet de spécifier les partiels afin d'obtenir un son de cloche (e.g. [partial 0.67 0.9 0.56 1] pour le second partiel cf. Chap. I 1.1.3). Chez J.-C. Risset il s'agit de l'association entre les sous-routines (PLF) et les « structures de cloche », qui lui permet de prescrire des sons de cloche d'une certaine complexité avec seulement trois paramètres.

35 « Whenever someone writes a program it is a program about something » [Winograd *et al*, 1994, p84]

4.3 Niveaux d'abstraction et information

Face à ce flou présent dans les niveaux d'abstraction et face à des niveaux relevant tantôt de l'informatique, tantôt de la cognition, voire de la perception (psychoacoustique), il est une théorie sur laquelle nous pouvons nous baser pour traverser ces niveaux indépendamment de toutes spécificités. Il s'agit de la théorie de l'information qui, en considérant les différences de quantité d'information, permet de comparer ces niveaux. Cette vue sur les artefacts, ou les DMIs, nous paraît assez courante. Nous donnons ci-après deux exemples, [Orlarey, 2009] et [Vinet, 2003], qui permettent d'appréhender ces différents niveaux avec une dimension scalaire.

Chez [Orlarey, 2009] trois « natures d'information à traiter » sont différenciées : (i) l'information du signal sonore, (ii) l'information « gestuelle ou performative » et (iii) l'information « conceptuelle », « plus abstraite », liée à la composition de l'œuvre³⁶. Une vue informationnelle permet ensuite de les traverser, de sorte qu'entre le niveau le plus bas et celui le plus haut il y ait d'une part réduction d'information, et de l'autre abstraction croissante.

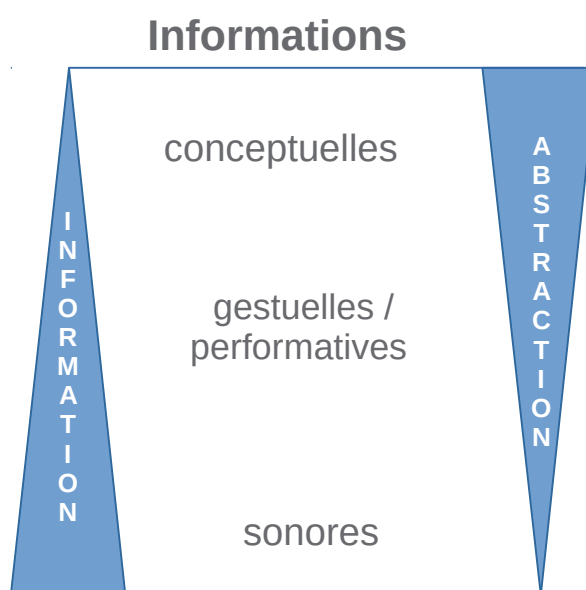


Fig. 5: trois natures d'information et leurs positionnements, d'après [Orlarey, 2009]

Chez [Vinet, 2003] la vue informationnelle est utilisée pour faire face aux difficultés de saisie des niveaux cognitifs et/ou perceptifs. Elle permet de mettre à plat ces derniers :

³⁶ Relevons que seul le deuxième niveau, « le gestuel », ne se recoupe pas avec les niveaux d'interaction de [Malloch et al., 2006], du signal au symbole.

Since these concepts [of musical information processing by the brains] are intended to modeling music representations in technical systems, and do not pretend to model music cognition, a more objective viewpoint for justifying this ordered organization can actually be found in information theory. *Ibid.*
p.197

En supplément des deux types de représentations que nous avons présentés (symbolique et signal), deux autres sont ajoutés dans l'article. Il s'agit « en bas » des représentations physiques ou spatiales, et « en haut » des représentations de connaissance. Les premières sont attachées au phénomène sonore, physique, et ici non pas considérées comme un signal (un point dans un espace) mais à l'inverse dans leur environnement spatial (« acoustic pressure as a function of space and time »). Les secondes concernent les indications textuelles présentes dans les partitions. Afin d'illustrer leur succession, une association est proposée entre ces quatre types de représentation et les niveaux impliqués dans la perception auditive (« processing stage of information auditory ») :

Niveaux de représentations musicales	Étapes de traitements de l'information auditive
Connaissance	Langage
Symbole	Connaissance des matériaux musicaux
Signal	Signal binaural
Physique	Forme du corps et position

Tab. 2 - Association hypothétique proposée dans [Vinet, 2003] entre des niveaux de représentation musicales et des étapes dans le traitement de l'information auditive.

Finalement, comme chez [Orlarey, 2009], à la verticalité de ces niveaux correspond évidemment une réduction de l'information.

Deuxième partie
Propositions

Chapitre III

Matériau musical et significations dans le système HAS

Résumé

Nous commençons ce chapitre en apportant de la lumière sur la notion riche de matériau musical, puis nous la confrontons au système HAS : comment le travail sur le matériau se fait-il au sein de ce système ? Nous utilisons le terme *paramètre* pour investir cette question et mettre en évidence un processus hautement dynamique, qui sous-tend l'élaboration du matériau. Ce processus dynamique est ensuite éclairé avec la théorie sémiotique de Charles Sanders Peirce. Enfin trois niveaux de structuration des artefacts sont définis.

1 Matériau musical

1.1 Introduction

Dans ce grand titre nous utilisons la notion de matériau musical dans le but d'investir la relation Homme-Son du système HAS. La lumière que nous apportons sur le matériau vient essentiellement d'un article, plutôt dense, de M.E. Duchez, *L'évolution scientifique de la notion de matériau musical*, [Duchez, 1991], qui permet de saisir certains aspects clés de cette notion. Parallèlement ce fil conducteur est enrichi par des écrits des compositeurs H. Lachenmann, H. Dufourt et enfin M. Chion. Il n'en reste pas moins que ce qui suit reste un éclaircissement et non un travail substantiel sur cette notion. Notre intention est de l'utiliser afin d'investir la relation Homme-Son, et non d'en rendre compte *in extenso*. Cet avertissement prend sens dans la mesure où le matériau est une notion des plus importantes dans la musique occidentale de la seconde moitié du XXe, comme le relève M. Solomos : « dans l'après 1945, tout se passe comme si [la notion de matériau] devenait centrale à la musique, allant même jusqu'à évincer d'autres notions » [Solomos, 2013, p.281]. Par voie de conséquence c'est aussi une notion qu'il n'est pas aisé de traiter – c'« est une notion des plus difficiles à manier » [Solomos, 1998] – dont la saisie mérite un effort certain. L'importance et la difficulté de cette notion se retrouve par exemple dans un écrit du compositeur J.-B. Barrière qui, en 1986, s'étonnait qu'une notion aussi récurrente en musicologie ait aussi peu été traitée scientifiquement (cf. [Barrière, 1986]). L'éclaircissement que nous proposons mériterait ainsi d'être approfondi ultérieurement avec des écrits de T.W. Adorno, P. Boulez, E. Varèse, ou encore G. Grisey.

1.2 Première approche : l'analogie

Au regard de ces circonspections il nous paraît opportun de commencer par une analogie avec un travail d'artisanat, celui du potier : le matériau est la glaise que le potier utilise pour donner une forme à son objet.

Le matériau serait alors ce avec quoi le compositeur compose, façonne son œuvre :

« dans le dualisme traditionnel matière-forme, [le matériau] est la matière dont le créateur tire une multitude de formes » [Duchez, 1991, p. 48]

Avec une vue volontairement schématique, le matériau serait, en musique instrumentale, l'ensemble des sons émanant des instruments traditionnels, qui sont prescriptibles avec la partition. Cette vue correspondrait à ce que Pierre Schaeffer a appelé les « sons notables » :

Une double limitation : l'une réservant aux sons « notables » les qualités de matériau musical, l'autre réservant aux œuvres « exécutable » – donc surtout à l'exécution – la qualité de manifestation musicale [Schaeffer, 1952, p. 144]

Il serait alors possible d'aller plus loin dans la compréhension du matériau en mettant en rapport le signe graphique de la partition avec le phénomène sonore produit. On pourrait alors se laisser tenter à analyser ce que le signe graphique, écrit par le compositeur, décrit du phénomène sonore. Alors le matériau serait approché avec une *description physique* du son, en termes de hauteur, amplitude, spectre et durée par exemple.

Une telle conception est par exemple présente, passagèrement, dans un article du compositeur Helmut Lachenmann, [Lachenmann, 1991]. Parmi les quatre aspects fondamentaux du matériau musical présenté, le deuxième est en effet « la *réalité physique* immédiate du phénomène sonore »³⁷. *Ibid.* p.266 (souligné dans le texte).

1.3 Son physique et son perçu

Cependant nombreux sont les auteurs qui nous mettent en garde contre cette volonté de saisir le matériau musical avec la réalité physique du son. Le matériau musical ne peut en effet se décrire objectivement avec des mesures physiques de phénomènes sonores. Relevons :

- « le matériau ne se réduit pas au son » [Barrière, 1986] ;
- « il n'existe aucun matériau qui ne se laisse décrire comme un phénomène purement physique » [Lachenmann, 1991, p. 266] ;
- ou encore M.-E. Duchez qui souligne « l'irréductibilité du son perçu par la sensibilité au son physique de l'acoustique » [Duchez, 1991, p. 49].

Le matériau musical est moins un son objectivé par les appareils de mesures physiques, ou encore par les signes graphiques de la théorie musicale, que « le son perçu, écouté, entendu » par le compositeur (*Ibid.* p. 49). Les mises en garde des auteurs nous invitent donc à effectuer un *distinguo* entre le son physique et le *son perçu*. Et si ces derniers sont inévitablement en relation, il faut remarquer que celle-ci n'a rien de linéaire. Par exemple, à un niveau psychoacoustique, cette non linéarité s'exprime par la différence qui réside entre les décibels et la sonie, ou encore entre la fréquence et la hauteur, ou plus largement avec les célèbres courbes isosoniques. Enfin, dans une perspective plus musicale nous pouvons paraphraser M.-E. Duchez : le son physique est le substrat du son perçu sans en être l'unique fondement (cf. [Duchez, 1991, p. 50]).

37 Le premier est la tonalité ; le troisième la structure et enfin le dernier l'aura.

1.4 Un son musical

Cette dernière distinction, entre son physique et son perçu, où le second ne peut se réduire au premier, soulève la question réciproque : qu'est-ce que le matériau musical a de plus que le son physique ?

Afin de répondre à cette question nous nous aidons d'un article de H. Dufourt intitulé *Logique du matériau (1970/1980, un bilan)* [Dufourt, 1980]. Dans cet article l'auteur avance que « le matériau sonore est devenu une nécessité théorique », puis donne trois raisons à cette nécessité. La première est qu'il existe une « scission totale » entre le musical et le sonore ; la seconde concerne les « prodigieux essors de la pratique instrumentale » ; la dernière, d'ordre esthétique, concerne le rejet du formalisme musical, « le formalisme musical ne rencontrait plus aucun consensus ».

C'est la première raison qui nous intéresse ici. L'auteur observe une scission entre le musical d'une part, où « les musiciens héritaient [en 1970] d'un formalisme, de loi combinatoire, de règle de syntaxe » aigu, et le sonore d'autre part, où les électroacousticiens « libéraient les puissances du son, en réabsorbant dans de larges trames le bruit, les sonorités complexes et le raz-de-marée des mixtures inharmoniques ». Le compositeur, de tradition résolument écrite, se trouve ainsi dans un certain dilemme :

« ou bien une écriture condamnée à fonctionner à vide, dans l'autarcie de ses systèmes formels ; ou bien une production sonore effective, mais frappée d'une intelligibilité de principe. (...) comment maîtriser les nouvelles sonorités par l'écriture ». *Ibid.*

Cet exemple nous permet d'apporter une réponse à notre précédente question. Ce que le matériau musical a de plus que le son physique se trouve dans le rapport au compositeur : le matériau est quelque chose avec lequel la musique peut se faire, quelque chose avec quoi le compositeur peut développer sa musique. Dans la précédente situation, « les puissances du son » libérées par les machines restent *sonores*, car elles ne possèdent pas de système d'écriture viable pour le compositeur, elles ne peuvent s'ajuster, rentrer en contact avec des développements musicaux. Le matériau « devient une nécessité théorique » car d'une part l'héritage formaliste – avec la syntaxe et ses lois combinatoires – n'ont pas d'emprise sur un « monde sonore en pleine effervescence », et de l'autre les « puissances du son » libérées par les machines ne peuvent être maîtrisées.

Le matériau ne peut ainsi se réduire à un son perçu puisqu'il est inévitablement en lien avec une volonté compositionnelle. En plus d'être un son perçu, le matériau est un son avec lequel le

compositeur possède des idées musicales. C'est un son perçu *dans une perspective de composition musicale*³⁸.

Dès lors, en plus des distinctions précédentes entre son physique et son perçu, il nous faut, avec M.-E. Duchez et *via* F. Bayle, en ajouter une supplémentaire, au plus proche de l'Homme, celle de *son musical* :

François Bayle affirme très justement que le passage nécessaire du son perçu au son musical – du matériau sonore, qui contient « dans sa gangue » le matériau musical potentiel, au matériau musical lui-même, « déjà perceptible et provocateur » dans le matériau sonore – est l'œuvre du musicien ; et que « la sélection » du matériau, « son travail » qui permettra de lui donner « valeur de signe », « sont partie intégrale et même considérable de la composition » [Duchez, 1991, p. 53] (les expressions entre guillemets proviennent d'un texte inédit de François Bayle ; nous soulignons)

Le son musical se différencie d'un son perçu dans la mesure où il a été travaillé. C'est un son qui coïncide avec des idées musicales, un son qui fait *valeur de signe* pour le compositeur.

1.5 Le matériau, dans l'interaction

Le son musical nous amène donc inévitablement à considérer le rôle du compositeur dans la notion de matériau. La voie alors la plus adaptée et la moins restrictive pour comprendre le matériau est de le concevoir au cœur de la relation Homme/Son. La notion de matériau ne prend sens ni totalement dans des phénomènes sonores qui se donneraient au compositeur, ni totalement dans des idées compositionnelles qui viendraient se transformer dans du sonore.

si l'on compose, c'est à la fois avec et contre le son (...). On doit donc accepter que le son conserve jusqu'au bout une certaine extériorité à la pensée compositionnelle [Chion, 1991, p. 69].

Avec M.-E. Duchez, qui cite Theodor W. Adorno, nous voyons que le moins restrictif et le plus fécond pour cette notion de matériau est de la concevoir avec d'innombrables interactions entre l'Homme et le Son :

Les directives que le matériau transmet au compositeur et que celui-ci transforme en leur obéissant se constituent dans une immanente interaction. T. Adorno³⁹ cité dans [Duchez, 1991, p. 53]⁴⁰.

38 On retrouve ceci chez Horacio Vaggione : « tout son utilisé en musique a déjà cessé d'être un simple son » [Vaggione, 1998].

1.6 Le matériau, dans le temps

En supplément de cette compréhension du matériau comme le fruit d'innombrables interactions, entre l'Homme et le Son, nous souhaitons en ajouter une dernière d'ordre temporelle. À l'issue des célèbres catégories du *hors-temps* et du *en-temps* amenées par Iannis Xenakis⁴¹, toutes deux présentes dans la composition, il paraît en effet nécessaire d'ajouter que le matériau est aussi ce avec quoi le compositeur a développé sa musique *dans le temps*, ce qu'il a utilisé pour établir des structures temporelles.

En effet si le concept de *son musical*, de son faisant « valeurs de signe », nous amène à fortement considérer les confrontations entre idées compositionnelles et le son perçu, il nous semble aussi important de pointer ce qui résulte de ces confrontations, c'est-à-dire les *choix* qui ont été faits par le compositeur, ce qui a été conservé, puis affiné, pour exposer des idées musicales.

Enfin, pour la suite de nos travaux, il nous est utile d'apporter une définition plus abstraite du matériau musical. Il correspondrait à un ensemble d'éléments, de quelque nature qu'ils soient, et de quelque quelle façon que l'on puisse définir ce terme « nature », qui d'une part seraient inévitablement en liens, de proche ou de loin, avec du sonore, et qui d'autre part auraient permis au compositeur de penser et de développer sa musique.

39 Theodor W. Adorno, *Philosophie de la nouvelle musique*, trad. H. Hildenbrand et A. Lindenberg, Paris, Gallimard, 1962, p. 45.

40 On peut alors comprendre l'article de M.-E. Duchez comme l'explication de ces interactions par deux dialectiques : celle du sujet et de l'objet et celle de l'abstrait et du concret. La première étant fortement nourrie par l'approche épistémologique de la musique de l'auteur, qui a travaillé sur la genèse de l'écriture musicale occidentale (e.g. [Duchez, 1981] et [Duchez, 1983]).

41 Cf. [Xenakis, 1963, p. 190-191] pour le texte original et [Solomos, 2008] pour une contextualisation.

2 Matériau et système HAS

Après avoir éclairé cette notion de matériau musical il nous faut à présent la questionner avec le système HAS. Si cette notion prend sens dans la relation Homme/Son, et si l'usage de l'informatique amène inévitablement à considérer une seconde relation, celle entre l'Homme et l'Artefact, qu'advient-il du travail sur le matériau musical ? Au sein du système HAS que se passe-t-il pour qu'un son puisse faire « valeur de signe » ?

2.1 Un corrélat

La première considération sur le matériau à apporter est qu'il se fait nécessairement avec une vue sur l'artefact. Au sein du système, dans l'acte compositionnel, à la perception du son correspond inévitablement une vue sur l'artefact. Le son n'est pas quelque chose de fini, mais il se pense, se saisit, avec ses moyens de productions. De façon très schématique si l'on peut comprendre que le matériau, dans la « composition traditionnelle », se perçoit avec des signes graphiques inscrits sur une page, avec l'informatique il se perçoit avec un ensemble de lignes alphanumériques. Soit, pour le dire de façon plus neutre, pour l'analyse de la musique d'informatique il paraît primordial de considérer que le son, sa perception, a pour corrélat quelque chose d'autre.

Chez [Bresson, 2007] par exemple, ce quelque chose d'autre tient en des représentations. Les techniques de synthèse, de même que leurs paramétrages (l'ensemble des valeurs que prennent les paramètres), sont des représentations du son :

chaque type de synthèse sonore met en avant une certaine représentation du son, définissant un domaine dans lequel un éventuel utilisateur évoluera pour l'observer et le manipuler. *Ibid.* p. 17.

Dans le précédent article de M.-E. Duchez les changements que pourraient apporter l'outil informatique sont aussi envisagés. La dernière partie de l'article traite en effet des technologies majeures du XXe : électroacoustique, électronique puis numérique. Pour l'auteur, après une lecture importante du T.O.M de P. Schaeffer, ce quelque chose d'autre tient en une « nouvelle causalité » :

Un procédé informatique défini possède, comme un instrument de musique, une permanence causale qui porte une information spécifique pour le compositeur ; on peut donc penser que, grâce à l'instauration d'une relation directe calcul/oreille humaine, une *nouvelle causalité* – abstraite mais prégnante et quasi immédiate, déjà ressentie mais non encore exprimée,

correspondant épistémologiquement (par une sorte de « geste intellectuel » devenu habituel) à la relation cognitive homme/instrument – se précisera dans l'ensemble conceptuel qui constitue la notion de matériau musical de la musique informatique. [Duchez, 1991, p. 67].

Chez [Orlarey, 2009] ce quelque chose d'autre tient en des processus d'engendrement :

la seule façon que nous avons d'imaginer un nouveau son, c'est de décrire le processus d'engendrement de ce son. *Ibid.* p. 358

Nous reprenons ce terme *engendrer* dans nos travaux puisqu'il laisse entendre l'aspect computationnel de l'ordinateur avec toutefois une certaine « distance »⁴².

Enfin, une compréhension plutôt limpide sur ces nouveaux éléments présents dans le travail compositionnel, se trouve dans des travaux non exclusivement musicaux, qui concernent le domaine du jeu vidéo. Chez [Farnell, 2007] il est ainsi question d'*audio procédural* : le son n'est pas quelque chose de fini, comme le pourrait être un fichier audio que l'on pourrait à loisir traiter, mais il est le fruit d'un processus d'engendrement, sur lequel on peut à loisir intervenir :

Procedural audio is sound qua process, as opposed to sound qua product. Behind this statement lies a veritable adventure into semiotics, mathematics, computer science, signal processing and music. (...). Procedural audio is non-linear, often synthetic sound, created in real time according to a set of programmatic rules and live input. *Ibid.* p. 1

[procedural audio] is not pre sequenced, pre-recorded sound and music *Ibid.* p. 12

L'ensemble des disciplines citées par l'auteur dans la première citation laisse entrevoir la richesse de saisie que l'artefact permet sur le son.

2.2 Problème de niveaux

Malgré cette relation computationnelle entre l'artefact et le son, l'entreprise qui viserait à établir un lien causal entre « ce quelque chose d'autre » et le matériau, nous paraît être dangereuse. En effet il nous semble important d'avoir en tête que le matériau musical du compositeur n'est pas forcément du même niveau d'abstraction que des paramètres présents

42 L'expression « l'artefact engendre la musique » nous paraît en effet moins restrictive que « l'artefact calcule la musique ».

dans l'artefact par exemple. La volonté d'éclaircir le matériau avec des éléments de l'artefact doit ainsi se faire avec une importante circonspection.

Ci-après nous utilisons le terme *paramètre* pour rendre compte, de façon très criante, de ce problème de niveaux. Nous mettons en lumière les différences de significations qui apparaissent entre (i) les paramètres du matériau de la musique sérielle avec (ii) les paramètres de l'artefact.

2.2.1 Paramètres dans la musique sérielle

Dans son article dédié au matériau musical, le compositeur [Lachenmann, 1991] témoigne qu'au moment de la période sérielle le matériau musical se pensait avec des paramètres. Ces derniers ne sont pas issus de la partition ou de la théorie musicale, mais plutôt des propriétés physiques du son :

[la réalité physique du matériau] s'est fait jour au cours de la dissolution progressive des catégories tonales opérées au plus tard par Webern, et après lui par les compositeurs sériels, chez qui le contrôle du matériau, issu à l'origine de la technique de variation et du développement, s'est finalement appliqué, de façon radicale, à toutes les propriétés acoustiques contrôlables du matériau : *ce qu'on a appelé ses « paramètres »*. [...]

Le premier stade de la mise à jour de [la réalité physique du matériau] a été la *pensée en paramètres* des compositeurs sériels. *Ibid.* p. 264 (souligné dans le texte)

Cette prégnance des paramètres dans la musique sérielle se retrouve aussi par exemple chez [Stroppa, 1999]. À cette époque la composition était « presque entièrement concentrée sur l'étude des propriétés du matériau abstrait », et la « réflexion compositionnelle » se faisait à l'aide de « paramètres contrôlant des matériaux abstraits » (p. 70).

Ces paramètres sont, pour l'essentiel, la hauteur, l'amplitude et la durée. On retrouve par exemple ces trois derniers dans la pièce historique *Studie II* (1954) de K. Stockhausen, pour son électronique fixé sur bande. En effet, une page de la partition⁴³ de cette pièce se divise verticalement en trois :

43 « La première partition de Musique Électronique publiée », d'après le texte introductif de la partition.

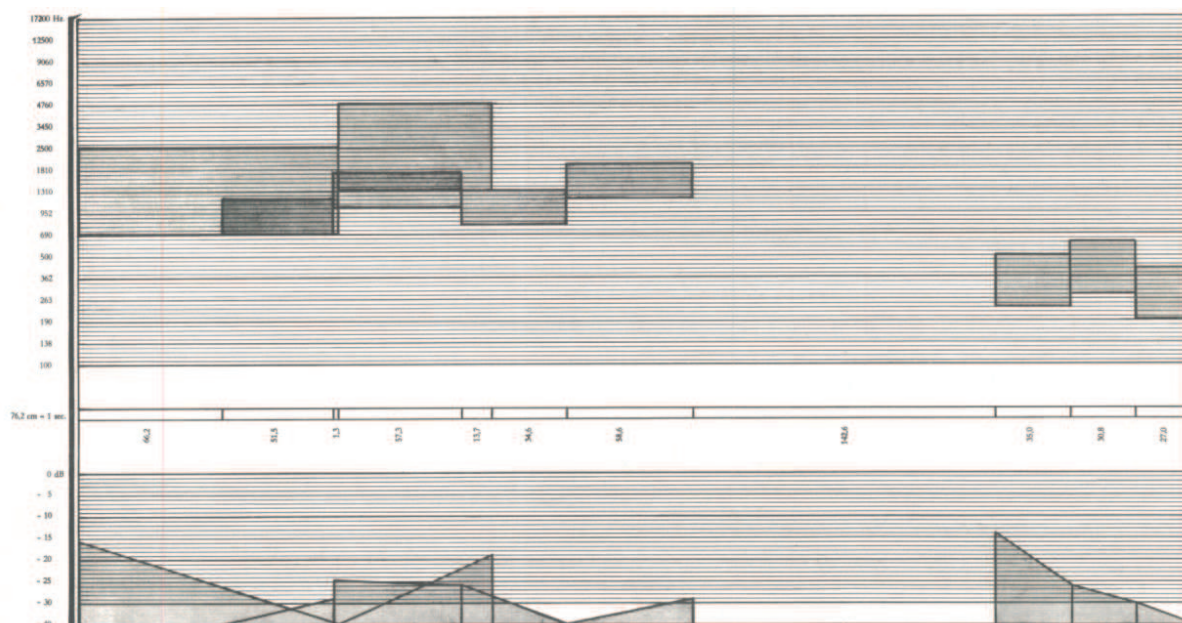


Fig. 6 - Première page de *Studie II* (1954), de K. Stockhausen.
 en haut l'espace des fréquences, au milieu celui du temps, en bas celui de l'amplitude
 © Universal Edition (London)

En haut se trouve l'espace des hauteurs, au milieu l'inscription temporelle du son, et en bas la dynamique, *i.e.* l'enveloppe d'amplitude des sons. De façon quelque peu simplifiée⁴⁴ on peut alors voir la composition comme la combinaison dans le temps de ces paramètres.

2.2.2 Paramètres dans l'artefact

Dans la mesure où les paramètres de l'artefact prennent sens avec le calcul, ou plus précisément avec les opérations mathématiques auxquelles ils sont reliés, ils s'avèrent *a priori* nettement différents des paramètres du matériau. On peut en effet comprendre que les racines des significations de ces deux termes ne sont pas les mêmes. Pour autant, les « paramètres de l'artefact » ne peuvent être totalement isolés du sonore, puisque l'artefact, tôt ou tard, peut être compris comme un tout produisant un signal. C'est peut-être pour cette raison que nous les retrouvons de façon très récurrente dans l'étude des DMIs (*e.g.* [Kvifte *et al.*, 2006]), ou encore qu'ils peuvent être envisagés comme des « représentations particulières » du son. [Bresson, 2007].

Nous proposons d'éclairer les paramètres de l'artefact en deux temps. D'abord en les considérant avec la fonction à laquelle ils se rattachent, et ensuite, plus fondamentalement, en

44 Notre simplification vient du fait que la composition ne s'est pas faite directement avec des sinusôides mais avec des complexes sonores. Chacun de ceux-ci est composé de 5 sinusôides – dont les valeurs en fréquence sont déduites d'un procédé mathématique – envoyées dans une grande réverbération d'environ 10 s. cf. [Manning, 2004, p. 49] et la partition éditée.

mettant l'emphase sur un recouvrement de significations qui a lieu dans le travail compositionnel.

2.2.2.a Signification computationnelle

En guise d'exemple prenons une simple ligne de code, dans le langage Csound, qui permet de générer une sinusoïde de 210 Hz :

```
asig oscils 0.2, 210, 0, 0
```

La syntaxe de la fonction *oscils* est la suivante : amplitude, fréquence, phase à l'origine et précision de calcul.

Cet exemple nous montre que les paramètres prennent sens avec les fonctions auxquelles ils se rattachent. Si des paramètres génériques sont possibles, comme la fréquence d'un oscillateur ou encore la fenêtre temporelle d'une analyse de Fourier, c'est-à-dire des paramètres qui prennent sens au-delà de la fonction et qui seraient communs à plusieurs langages, ces derniers restent inévitablement minimes en comparaison de la densité des artefacts. Ainsi, dans le soubassement de l'activité qui consiste à comprendre un artefact, se trouve l'intégration des significations qui sont figées dans le *référentiel* du langage utilisé, comme en témoigne, de façon minimale, la syntaxe de la fonction *oscils*.

On retrouve cette disparité des significations des paramètres chez J.-B. Barrière par exemple :

Les processus de description en synthèse ne sont pas unifiés : on spécifie des paramètres dont le nombre et l'effet diffèrent drastiquement d'un logiciel à l'autre. [Barrière, 1993, p. 11].

Nous utilisons le terme *computationnel* pour qualifier cette source de signification ; la *signification computationnelle des paramètres* qualifie ainsi le sens que prennent les paramètres selon les calculs qu'ils effectuent.

2.2.2.b Recouvrement de significations

Évidemment toutes les significations des paramètres présents dans un artefact ne peuvent être acquises par une simple connaissance du référentiel du langage. Comme nous l'avons vu brièvement au premier chapitre, les paramètres prennent aussi sens dans les connexions qu'ils entretiennent (cf. Chap. I 2.2). Afin d'illustrer ce second aspect de signification, nous présentons un simple exemple d'un vibrato sinusoïdal.

Soit un oscillateur sinusoïdal de fréquence 2 Hz calibré de sorte à osciller dans l'intervalle [-20, 20]. À ce signal est ajoutée une constante de valeur de 100. Ce nouveau signal est relié dans

l'entrée en fréquence d'un second oscillateur sinusoïdal, de pleine amplitude. Ce procédé présente donc un simple vibrato, que nous pouvons représenter de façon abstraite comme suit :

```
osc(2, 20) // un oscillateur de fréquence 2 et amplitude 20
+ 100
osc(in, 1) // in étant le signal produit à la ligne précédente
output(in)
```

Le paramètre d'amplitude du premier oscillateur est significatif au niveau de son oscillateur : il définit l'amplitude du signal résultant. Cependant, au niveau des deux oscillateurs, si on fait varier la valeur de ce paramètre, alors un problème de signification apparaît : ce paramètre d'*amplitude* ne fait plus sens vis-à-vis du résultat puisqu'il impacte désormais une déviation en *fréquence* : il impacte la hauteur du son produit.

Cet exemple simple – que les designers voudront bien nous excuser – nous montre que la signification des paramètres se fait par recouvrement. Le sens des paramètres déborde du référentiel d'un langage pour s'établir dans les connexions, ou plutôt dans le réseau de connexions réalisé lors de l'activité de *design*.

2.3 Matériau et jaillissement de significations

Plus substantiellement le terme *paramètre* nous montre que le matériau musical se conçoit, en informatique, avec un jaillissement de significations. Si évidemment ce matériau prend sens dans l'écoute du son, dans la relation Homme/Son, il nous faut, de plus, en musique d'informatique, considérer des significations qui viennent de l'artefact. Le matériau peut alors se comprendre comme quelque chose de construit, avec en effet un ensemble de paramètres, mais dont la signification vient de la rencontre entre (i) des calculs réalisés et (ii) de la perception du résultat sonore.

À la suite de M.-E. Duchez qui soulignait que le matériau se concevait dans la dialectique compositeur/son – et avec T. W. Adorno qu'il soit le fruit « d'immanentes interactions » –, il nous faut considérer qu'avec l'informatique ces interactions ont *aussi* lieu avec l'artefact.

Homme

(vue, action, écriture) (perception)



*Fig. 7-Le système HAS, au sein duquel
le matériau se conçoit*

Ainsi, en reprenant l'opposition « paramètres sériels » et « paramètres de l'artefact », nous voyons que le problème des premiers pour attaquer le matériau à l'heure de la musique d'informatique se trouve dans le fait que les significations sous-jacentes au matériau débordent de la relation Homme/Son ; elles prennent aussi place dans la relation Homme/Artefact.

Cette difficulté d'adaptation des « paramètres sériels » est par exemple présente chez M. Stroppa :

En fait, c'est le concept de paramètre qui doit céder sa place à une caractérisation *dynamique* plus adéquate des attributs d'un matériau [Stroppa, 1999, p. 70] (nous soulignons)

Ce jaillissement de significations au sein du système HAS nous amène *de facto* à considérer un aspect fortement *dynamique* du travail compositionnel. Les significations qui participent à l'élaboration du matériau sont d'une part issues de la rencontre entre l'artefact et le son, et d'autre part sont en constante évolution au cours du travail compositionnel.

3 Vers une caractérisation dynamique

3.1 Charles Sanders Peirce et la semiosis

Afin d'investir ces jaillissements de significations nous proposons d'utiliser la théorie sémiotique du philosophe américain Charles Sanders Peirce (1839-1914), à laquelle nous nous référons avec le terme *semiosis*⁴⁵. Nous précédon la présentation théorique d'une courte introduction.

3.1.1 Introduction

Les écrits du philosophe Ch. S. Peirce ont déjà inspiré des travaux musicologiques. Par exemple, pour n'en citer qu'un, la célèbre sémiologie⁴⁶ musicale essaimée par [Nattiez, 1987] n'a d'autre pour axiome « la conception dynamique du signe de Peirce », et, « tout le reste [du célèbre ouvrage de J.-J. Nattiez] est une défense et illustration de cette conception Peircienne du signe » (*Ibid.* p. 30). C'est ainsi que dans l'introduction du célèbre ouvrage de J.-J. Nattiez nous retrouvons, à la suite d'une excursion dans la sémiologie de Ferdinand de Saussure, une exposition de cette conception dynamique du signe.

Avant d'aller plus loin avec la sémiotique de Peirce il nous paraît important de prendre des précautions sur ce que nous pouvons en dire. En effet, pour reprendre J.-J. Nattiez, ce que nous disons de cette sémiotique reste fondamentalement « *une* interprétation et *une* utilisation possibles » (*ibid.* p. 29, soulignés dans le texte).

En amont de la question de l'interprétation, il nous semble que ces précautions peuvent s'expliquer par deux points. D'abord, du vivant du philosophe, aucune monographie n'a été publiée sur ces réflexions sémiotiques :

les deux seuls ouvrages de Peirce qui parurent de son vivant sont un livre d'astronomie : *Photometric Researches* (1978) et un recueil d'études de logique qu'il publia en collaboration avec ses étudiants pendant la brève période où il enseigna : *Studies in Logic* (1883). [Deledalle, 1978, p. 8]

C'est donc dans une publication posthume que les réflexions du philosophe se sont fait essentiellement connaître. Cette publication n'est ainsi pas une monographie, structurée, mais

45 « Le processus selon lequel quelque chose fonctionne comme un signe peut être appelé *semiosis* ». Citation présente dans [Nattiez, 1938, p. 61], de C. Morris, 1938, « Foundations of the Theory of Signs », International Encyclopedia of Unified Sciences, vol 1, n° 2, Univ. of Chicago.

46 « En bonne morale terminologique, la sémiotique est la théorie Peircienne des signes et la sémiologie la théorie saussurienne des signes » [Deledalle, 1978, p. 212].

un recueil de ses écrits philosophiques, nécessitant pas moins de huit volumes : *Collected Papers*⁴⁷. Ensuite – et, comme le suggère [Deledalle, 1978, p. 10], cela pourrait être une explication de cette absence d'édition de monographie – les écrits du philosophe ne sont pas des modèles d'une réflexion cohérente et limpide :

Peirce était un penseur profond (...) rigoureux et exigeant et en même temps un écrivain expérimentaliste qui suivait, pour voir, le fil d'un raisonnement jusqu'aux limites du soutenable, refusant obstinément de se laisser enfermer dans le carcan d'un sujet et faisant fi de toutes les règles de genre. *Ibid.* p. 10.

À titre d'exemple J.-J. Nattiez précise qu'il a pu relever dans une partie des *Collected Papers*, « pas moins de douze définitions différentes du signe et de l'interprétant » [Nattiez, 1987, p. 28].

C'est donc évidemment avec ces précautions que nous nous permettons de présenter un fragment de la théorie sémiotique du philosophe. Notre lecture vient essentiellement de [Deledalle, 1978], qui a rassemblé, traduit et commenté des textes du recueil *Collected Papers*.

3.1.2 La semiosis

La semiosis de Peirce se manifeste au sein d'une triade, impliquant un premier, un *representamen*, un second, un *objet*, et un troisième, un *interprétant* :

Un representamen est le sujet d'une relation triadique avec un second appelé son *objet*, pour un troisième appelé son *interprétant*, cette relation triadique étant telle que le *representamen* détermine son interprétant à entretenir la même relation triadique avec le même objet pour quelque interprétant (1.541)⁴⁸ [Deledalle, 1978, p. 117].

Afin de guider le lecteur néophyte, de lui éviter de fausses directions, apportons ci-après des considérations sur chacun des éléments participant à la semiosis :

- l'*interprétant* n'est pas l'interprète, il n'est pas une personne qui reçoit « par la porte des perceptions » un signe. Il est un *moyen* qu'une personne « utilise pour effectuer son interprétation » [Everaert-Desmedt, 1990] ;

47 Peirce, C. S., *Collected Papers of Charles Sanders Peirce*, Harvard University Press, 1931-1935, puis 1958.

48 Les citations qui viennent directement du recueil sont suivies d'un numéro, ici 1.541 – qui indique le numéro du volume et celui du paragraphe –, et l'absence de ce numéro signifie qu'il s'agit d'un commentaire de G. Deledalle.

- le *representamen* n'est pas quelque chose de matériel : il ne peut se réduire à une inscription graphique sur du papier, il ne peut se réduire à la perception d'un phénomène sonore ;
- l'*objet* non plus n'a rien de matériel, il « est un « incorporel », de quelque manière qu'on définisse cet incorporel » [Deledalle, 1978, p. 216].

Ces considérations soulignent que le processus de signification chez Peirce est résolument *abstrait* : il s'applique indifféremment aux choses entrant par la porte des perceptions. Comme le souligne [Barthes, 1964], renvoyant lui-même à Saint Augustin, le terme *signe* se conçoit dès lors qu'il met en relation deux choses⁴⁹, et peu importe la nature de ces choses.

En concevant ces renvois, cette relation entre choses amenées par le signe, il nous semble que l'on peut éclairer la sémiotique de Peirce par le fait (i) qu'elle instaure deux fois une telle relation, et (ii) que toutes deux s'articulent par l'intermédiaire d'un élément *ad hoc*, l'*objet*, de sorte à former une triade.

Le signe est en effet présent :

- dans la relation *representamen / objet* :

Mais pour que quelque chose soit un signe, il faut, comme on dit, qu'il « représente » quelque chose d'autre, appelé son *objet*. (2.230) [Deledalle, 1978, p. 122].

- dans la relation *interprétant / objet* :

L'interprétant d'un signe est « un second signe qui est déterminé par le signe primitif à renvoyer au même objet auquel le signe primitif renvoie et qui traduit la signification du signe primitif ». *Ibid.* p. 222

Ainsi le signe est, au cours du processus de signification, tantôt le *representamen* – « un premier, une pure potentialité » [Everaert-Desmedt, 1990] – tantôt l'*interprétant* – « celui qui opère la médiation entre le *representamen* (premier) et l'*objet* (second) » (*Ibid.*) :

un signe, ou *representamen*, est quelque chose qui tient lieu pour quelqu'un de quelque chose sous quelque rapport ou à quelque titre. Il s'adresse à

49 « une *relation* entre deux *relatas* » [Barthes, 1964, p. 103] (souligné dans le texte) Puis R. Barthes ajoute en note de bas de page : « Ce qu'a exprimé très clairement Saint Augustin : « Un signe est une chose qui, outre l'espèce ingérée par les sens, fait venir d'elle-même à la pensée quelque autre chose » ».

quelqu'un, c'est-à-dire crée dans l'esprit de cette personne un signe équivalent ou peut être un signe plus développé. Ce signe qu'il crée, je l'appelle l'*interprétant* du premier signe. Ce signe tient lieu de quelque chose : de son *objet*. (2.228) [Deledalle, 1978, p. 121].

3.1.3 Dynamique du signe

Dans cette présentation trop succincte, la perspective la plus féconde pour nos travaux – sur laquelle se base aussi la sémiologie de Nattiez⁵⁰ –, se trouve dans l'aspect *dynamique* de la signification. Cet aspect est présent d'abord dans le fait que l'*interprétant* est lui-même un signe d'un premier signe, le *representamen*, et ensuite par le fait qu'un *interprétant* devienne, au fil du processus, un *representamen*, et ainsi de suite.

Le *representamen*, pris en considération par un interprète, a le pouvoir de déclencher un *interprétant*, qui est un *representamen* à son tour et renvoie, par l'intermédiaire d'un autre *interprétant*, au même objet que le premier *representamen*, permettant ainsi à ce premier de renvoyer à l'objet. Et ainsi de suite à l'infini. [Everaert-Desmedt, 1990].

La signification passe donc par l'intégration de nombreux signes, par la série *representamen-interprétant*, par l'établissement de signe du signe :

[en bref, un signe est] tout ce qui détermine quelque chose d'autre (son *interprétant*) à renvoyer à un objet auquel lui-même renvoie (son *objet*) de la même manière, l'*interprétant* devenant à son tour un signe et ainsi de suite *ad infinitum*. (2.203) [Deledalle, 1978, p. 126].

un signe doit pouvoir toujours renvoyer à un signe. *Ibid.* p. 220

3.2 Des significations dans l'artefact

Nous appliquons maintenant cette théorie aux deux puits de significations présents dans le système HAS, dans la relation Homme/Artefact et celle Homme/Son.

50 À maintes reprises, dans l'introduction, il est question de *la conception dynamique du signe Peircien*, et à celle-ci est opposé celle *statique* de F. de Saussure [Nattiez, 1987, p. 30].

3.2.1 Deux processus de signification ?

3.2.1.a Homme/Artefact

Nous avons vu qu'une première signification des paramètres était figée dans le référentiel du langage de programmation, c'est ce que nous avons appelé la *signification computationnelle*. Celle-ci est figée puisqu'elle n'appartient pas au travail compositionnel mais au travail d'implémentation du langage, réalisé par des ingénieurs/chercheurs/designers.

Le travail compositionnel commence avec cet état initial de significations, puis intègre des fonctions et paramètres dans ce qui va devenir l'artefact. Les paramètres changent alors de signification selon les connexions réalisées, comme nous l'avons montré avec l'exemple du vibrato sinusoïdal.

Dans ce travail, on peut voir les significations computationnelles des paramètres, figées, comme des *representamen*. Ensuite, l'intégration de ces derniers dans l'artefact, par l'établissement d'un réseau de connexions, dense, donne lieu à une succession de *representamen-interprétant*, et ainsi de suite, durant tout le travail compositionnel. Cette succession ne peut avoir lieu qu'avec l'*objet* Peircien, que nous voyons ici, en tant qu'incorporel, qui ne peut qu'être représenté, comme les idées du compositeur.

3.2.1.b Homme/Son

Dans la relation Homme/Son les signes sont évidemment moins saisissables puisqu'ils ne laissent pas de traces. L'éclaircie que nous avons faite sur la notion de matériau peut ici nous être utile. En fait, même si cela n'est pas présent de façon explicite, la théorie du signe dans l'article de M.-E. Duchez ([Duchez, 1991]) reste, selon nous, latente. C'est par exemple limpide dans la distinction entre un *son perçu* et un *son musical*, où le second fait « valeur de signe »⁵¹ pour la composition, c'est-à-dire qu'il possède, pour le compositeur, un poids significatif nettement plus fort que le premier, ou encore qu'il s'assimile avec l'intention compositionnelle.

C'est à l'aide de ces deux niveaux que nous appliquons la semiosis. Entre le passage du *son perçu* au *son musical* correspond un jaillissement très dense de *representamen-interprétant*, qui se fait par le travail compositionnel. Ce jaillissement est très disparate pour le *son perçu*, puis,

⁵¹ Cette expression est issue d'un entretien avec le compositeur F. Bayle (cf. [Duchez, 1991, p. 53]). Elle retient particulièrement notre attention puisque F. Bayle a investi l'écoute – avec le concept connu de *i-son* – en faisant des parallèles avec une théorie triadique de Peirce (cf. [Bayle, 1989]). Cependant nous ne rendons pas compte de ce parallèle puisque la triade utilisée par F. Bayle, philosophique, n'est pas la même que celle que nous avons présentée, sémiotique. Toutes deux restent néanmoins complémentaires, si ce n'est imbriquées. Disons simplement ici que la triade philosophique de Peirce amène trois catégories, la *priméité*, la *secondéité* et la *tiércéité*, et que les éléments de la semiosis (*representamen*, *interprétant* et *objet*) se subdivisent eux-mêmes en trois selon ces catégories. Ainsi, après avoir témoigné des distinctions schaefferiennes entre ouïr, entendre et écouter, F. Bayle utilise les catégories philosophiques de Peirce pour investir la profondeur de champ de l'écoute (cf. [Bayle, 1989], et [Bayle, 2006]).

par le travail, par l'action du compositeur sur ce qu'il écoute, ces successions de *representamen-interprétant* vont comme être canalisées jusqu'à ce qu'il y ait une sorte de concordance entre ce qui est écouté et des intentions musicales, jusqu'à ce que le son devienne *musical*. Dans cette succession de *representamen-interprétant* nous envisageons de nouveau l'*objet* Peircien comme des idées compositionnelles.

3.2.2 Rencontre de significations

À la suite de cette application il serait tentant de voir deux processus de significations distincts. L'un à partir de « *representamen* computationnels », les paramètres, et l'autre à partir de « *representamen* sonore ».

Cependant, du fait que les *representamen* ne peuvent se réduire à quelque chose de matériel – ce qui revient à dire au fond que la signification siège dans la cognition de l'Homme – nous ne pouvons opposer deux objets Peircien selon une différence de « nature ». L'aspect *dynamique* de la sémiotique nous appelle bien plus à voir un jaillissement de *representamen-interprétant*, très foisonnant, qui peut, éventuellement, se faire au début du travail compositionnel avec des *objets* divergents mais qui, par l'action du travail compositionnel, viendrait se rapprocher d'un seul et même *objet*, aussi polymorphe soit-il, qui correspondrait aux idées compositionnelles.

La pluralité des signes, parfois computationnels, parfois perceptifs, dans le travail compositionnel peut être mise en correspondance dans la théorie de Peirce avec l'existence d'*objets complexes*. L'exemple suivant permet d'illustrer ce principe de « non unicité des *objets* » :

Un signe peut avoir plus d'un objet. Ainsi la phrase « Caïn tua Abel » qui est un signe, renvoie au moins autant à Abel qu'à Caïn, même si elle n'est pas considérée comme il le faudrait, à savoir comme ayant un « assassinat » comme troisième objet. Mais l'ensemble des objets peut être regardé comme formant un objet complexe. (2.230) [Deledalle, 1978, p. 123].

La non concordance entre un signe computationnel et un signe sonore, issue de la perception, pourrait alors s'expliquer comme un signe qui représenterait autre chose que son objet. Nous trouvons ce cas chez Peirce. Il se comprend par l'aspect dynamique de la théorie, que nous voyons comme une imbrication de signes :

Si un signe est autre que son objet, il doit exister, soit dans la pensée, soit dans l'expression, quelque explication ou argument ou quelque contexte montrant comment – dans quel système ou pour quelle raison le signe représente l'objet ou l'ensemble des objets qu'il représente. Or, le signe et l'explication forment

un autre signe et puisque l'explication sera un signe, elle requerra probablement une autre explication qui, ajoutée au signe déjà plus étendu, formera un signe encore plus vaste ; et en continuant ainsi nous parviendrons ou devrions parvenir en fin de compte à un signe qui est signe de lui-même, contenant sa propre explication et les explications de toutes ses parties significatives ; et suivant cette explication chacune de ces parties a quelque autre partie pour objet. (2.230) *Ibid.* p. 123

Ainsi, dans le travail compositionnel nous ne voyons pas deux processus distincts, mais bien un seul. Celui-ci est fortement dynamique, et à cet aspect dynamique correspond un *objet* qui peut être vu comme résolument complexe. Cette complexité serait alors extrêmement présente au début du travail, lors des premiers jaillissements de signes, qui correspondent aux premières représentations des idées musicales. Puis, dans le temps, dans l'acte compositionnel, par une succession incommensurable de *representamen-interprétant*, d'imbrications de signes, cette complexité se façonnerait, s'amoinrirait peu à peu.

Au travail compositionnel correspond donc l'intégration de significations, avec des significations d'engendrement (par la voie computationnelle) d'une part, et des significations perceptives (par l'écoute) de l'autre, jusqu'à ce que le rendu fasse « valeur de signe », c'est-à-dire soit approprié pour développer des idées musicales.

3.2.3 Variables, paramètres et contrôles

Il nous semble que cette intégration de significations peut s'illustrer par les différences entretenues entre trois termes devenus quasi inévitables en analyse comme en composition avec l'outil informatique : *variables*, *paramètres* et *contrôles*. Ces derniers sont en effet aussi bien présents dans des travaux techniques (e.g. [Puckette, 2005]), que compositionnels (e.g. [Stroppa, 1999]), que musicologiques (e.g. [Pottier, 2011]).

À l'issue de notre parcours dans la sémiotique de Peirce, nous apportons ci-après des éclaircies sur chacun de ces termes clés. Ceci doit permettre de distinguer trois *moments* dans l'acte compositionnel, et par là même trois niveaux dans l'artefact.

3.2.3.a Variables

Dans le processus compositionnel une variable est un premier, dans le sens où le compositeur lui accède sans qu'elle n'ait *a priori* de signification musicale. C'est quelque chose qui appartient au langage de programmation que le compositeur va utiliser pour effectuer un lien.

La variable tire davantage sa signification dans le calcul qui est effectué – voir dans le langage de programmation – que dans le travail compositionnel à proprement parler. Le sens vient par exemple des conventions du langage, où les variables se différencient selon leurs *typages* :

variable de contrôle, variable audio, variable de nombre entier, variable de nombre flottant, variable de caractère. La variable est donc de « bas niveau ». Son sens ne vient pas de l'usage musical qui en est fait. Elle est, par là même, générique, elle transcende toutes les fonctions computationnelles.

3.2.3.b Paramètres

Les paramètres prennent eux leur sens avec le travail compositionnel. Ils ne sont plus en lien avec les conventions, ou le référentiel du langage de programmation, mais ils viennent qualifier ce que fait quelque chose qui a été assimilé par le compositeur.

Cette chose peut être simplement une fonction du langage, comme une fonction de délai. Le paramètre hérite alors d'une signification computationnelle, comme nous l'avons vu.

Cette chose peut aussi être plus conséquente, un fragment de l'artefact réalisé par le compositeur, comme les sous-routines de Music N ou les abstractions de Pure Data. La signification du paramètre vient alors du réseau de connexions établies entre fonctions et paramètres.

Cette chose peut aussi engendrer, paramétrer, un signal audionumérique. La signification du paramètre se fera alors entre l'assimilation de la computation réalisée et la perception du signal, du *son perçu*.

Les *paramètres* héritent donc de leurs significations au cours du travail compositionnel, auquel correspond le processus de signification fortement dynamique. À la différence des *variables*, le moment, le contexte, à partir duquel ils prennent sens, reste très flou, difficilement circonscriptible.

3.2.3.c Contrôles

Les *contrôles*, ou *paramètres de contrôle* parfois, permettent de sortir du flou de signification qui réside dans les paramètres. Si ces derniers peuvent se situer à l'intersection des deux puits de significations, computationnel et sonore, les contrôles eux se situent au-dessus, ils en résultent, de sorte que leurs significations s'établissent à un niveau musical.

Nous pouvons par exemple rendre compte de ce terme dans un travail sur les DMIs. Dans [Birnbaum *et al.*, 2005] les auteurs proposent un espace dimensionnel avec lequel représenter, saisir, analyser ces dispositifs. L'espace est formé par 7 dimensions, dont l'une concerne le contrôle musical⁵² :

⁵² Les six autres dimensions sont : « distribution in Space, Role of sound, Required Expertise, Degrees of Freedom, Feedback Modalities, Inter-Actors »

The Musical Control axis specifies the level of control a user exerts over the resulting musical output of the system. *Ibid.* p. 193

De façon plutôt étonnante cet axe n'est pas continu, mais constitué de trois points discrets⁵³ : le premier concerne un niveau timbral, le second le niveau de la note, et le dernier, inévitablement plus flou, concerne les « contrôles sur les processus musicaux » (« control over a musical process »).

Si la signification des paramètres peut être autre que musicale, c'est-à-dire fortement computationnelle, mathématique, celles des *contrôles* l'est résolument. Les contrôles sont issus du travail compositionnel. Nous les voyons comme des *points d'accès* que le compositeur s'est construit et va utiliser pour développer sa musique.

Dans la vue Peircienne, où l'on a assimilé l'*objet complexe* à un ensemble d'idées musicales, les *contrôles* sont alors des signes de ces idées, ils représentent cet *objet complexe*. Ils sont ainsi au plus proche de la cognition du compositeur ; c'est avec eux que la musique s'est pensée, développée. Ils sont les éléments de l'artefact qui ont catalysé le plus de significations. Comme nous le verrons au chapitre suivant, ce sont vers l'éclaircissement de ces derniers que l'analyse doit tendre.

53 Les auteurs précisent que cette classification est reprise de l'article suivant : W. Andrew Schloss. « Recent advances in the coupling of the language Max with the Mathews/Boie Radio Drum ». In *Proceedings of the International Computer Music Conference*, pages 3 98–400, 1990

Chapitre IV

Vers un cadre conceptuel pour la description des artefacts

As the use of a new technology changes human practices, our ways of speaking about that technology change our language and our understanding.

Terry Winograd and Fernando Flores

Résumé

Le cœur de ce chapitre est la proposition d'un cadre conceptuel pour organiser la description des artefacts. La définition du cadre proposé est précédée d'une analogie avec un autre cadre conceptuel usuel dans un autre domaine, celui des bibliothèques où la description bibliographique est centrale. Dans un second temps nous éclairons l'aspect dynamique de notre cadre avec une paire de concepts présente chez M. Stroppa, nommée *potentiel* et *actualisation*. Nous enrichissons ces derniers avec deux autres paires de concepts, célèbres dans l'étude du langage, *langue* et *parole* (Saussure) et *compétence* et *performance* (Chomsky). Enfin nous concluons avec le concept de *structure de contrôle*, qui donne la première réponse à notre problématique : l'analyse musicale peut s'aider de l'artefact en mettant au jour sa *structure de contrôle*.

Introduction

Dans le précédent chapitre nous avons montré qu'au travail compositionnel correspond un jaillissement très dense de significations. Corrélativement, on peut d'ores et déjà comprendre que le travail de l'analyse doit aussi passer par l'intégration de significations. Une quête pour l'analyse est dès lors de viser les contrôles de l'artefact, les significations que les contrôles ont agrégées.

Afin d'aider l'analyste nous proposons ici un cadre conceptuel, un ensemble d'idées, de concepts, qui doit aider à « remonter » aux contrôles de l'artefact. Ce cadre a pour volonté de faciliter la description des artefacts. L'enjeu pour l'analyste est de mettre à jour une certaine structuration, de sorte que la communauté puisse savoir, en minimisant au maximum les polysémies, quelles sont les idées fortes que l'artefact véhicule.

Ce cadre est essentiellement issu de notre pratique de l'analyse de la pièce *Jupiter* de Philippe Manoury. Il s'agit donc résolument d'une proposition. Ce serait une erreur de le comprendre « par le haut », comme un ensemble conceptuel à appliquer tel quel. Il n'est donc pas impossible que notre terminologie, à l'issue d'autres applications, soit à revoir.

En préambule de notre proposition nous effectuons une analogie avec un autre modèle conceptuel très utilisé pour les descriptions bibliographiques, le « modèle FRBR ».

1 Analogie avec un modèle bibliographique

Nous commençons la description du cadre conceptuel par une analogie avec un autre cadre, ou modèle, présent dans la description des ressources bibliographiques. Ce modèle est nommé « Fonctionnal Requirement Bibliographic Record », ou plus communément sous sa forme abrégée, « modèle FRBR ».

La compréhension des enjeux de ce modèle, riche d'une vingtaine d'années⁵⁴, doit nous permettre de mettre en lumière, par analogie, les enjeux de la définition d'un modèle adapté à notre contexte. C'est parce qu'il est difficile de voir de prime abord les aboutissants d'un modèle conceptuel que nous proposons cette analogie.

1.1 Vue sommaire du modèle FRBR

1.1.1 Contexte d'émergence

Le modèle FRBR a été conçu dans les années 1990 à la suite de travaux sur la description des ressources bibliographiques, organisés par une fédération internationale, l'IFLA⁵⁵. À cette date une norme internationale de description bibliographique⁵⁶ existait déjà, l'ISBD⁵⁷, qui fut réalisée une vingtaine d'années auparavant. Le besoin de retravailler sur les descriptions bibliographiques naquit d'un éventail de faits parmi lesquels un retient particulièrement notre attention : l'hétérogénéité des supports auxquels sont confrontées – 20 ans après l'établissement de la dernière norme internationale – les institutions qui réalisent les descriptions. La réalisation du modèle se fait ainsi dans une volonté de dépasser cette diversité :

L'étude [sur les descriptions bibliographiques à échelle internationale] cherche également à couvrir le plus grand nombre possible de types de documents, de supports et de format. [Leresche *et al.*, 2012, p.9].

Parallèlement, la réalisation du modèle s'est faite dans la volonté de répondre, autant que possible, aux différents *usages* des notices bibliographiques :

L'étude suit une méthode qui met l'accent sur l'utilisateur pour analyser les spécifications de données dans la mesure où elle cherche à définir de manière

54 Le modèle fut créée en 1997, et il est toujours d'actualité.

55 International Federation of Library Associations.

56 Une norme de description bibliographique précise formellement par exemple comment référencer un livre publié. Elle contient différents champs (*e.g.* titre, auteur, ISBN, éditeur, collection, pagination) et précise le type de données que chaque champ doit accueillir.

57 International Standard Bibliographic Description.

systematique ce sur quoi l'utilisateur s'attend à être renseigné lorsqu'il consulte une notice bibliographique. *Ibid.* p. 8.

En synthèse retenons donc que ce modèle met en relation (i) les données présentes dans les notices bibliographiques, avec (ii) les besoins des utilisateurs.

1.1.2 Définition du modèle

Il ne sera pas question ici de définir le modèle *in extenso*, ce qui nécessiterait un développement substantiel, mais simplement de donner un aperçu de sa structure globale. Le modèle se construit avec une terminologie d'une dizaine de mots – indiquée en italique ci-après –, et la densité du modèle tient justement dans la définition de chacun de ces termes. Le modèle peut ainsi être vu comme l'établissement d'une terminologie qui doit permettre de se référer aux éléments structurants du domaine bibliographique, sans qu'il y ait confusion.

Les premières étapes nécessaires à la définition du modèle consistent à identifier des *entités*, à la fois présentes dans les notices et fondamentales pour les utilisateurs. Ensuite, il est nécessaire de relever des *relations* présentes entre les différentes *entités*. Enfin il faudra relever les *caractéristiques* (ou *attributs*) des *entités*.

La suite de la définition du modèle, que nous n'aborderons pas ici, consiste à confronter cette structure globale « entités, relations, attributs » avec quatre types d'usages relevés dans l'utilisation des notices.

1.1.3 Exemple

Afin d'illustrer le modèle nous donnons ci-après un exemple d'*entités*, puis un exemple courant présent dans l'édition bibliographique.

Les quatre entités les plus connues sont l'*Œuvre*, qui se réalise dans une *Expression*, qui elle-même se matérialise dans une *Manifestation*, qui elle-même a pour exemplaire un *Item*.

Ces quatre entités se différencient deux à deux à l'aide d'un critère de matérialité (cf. ([Leresche *et al.*, 2012, p. 100])) :

- l'*Œuvre* et l'*Expression* expriment le *contenu intellectuel* ou *artistique* ;
- la *Manifestation* et l'*Item* expriment la *forme matérielle* du document.

Le schéma ci-après permet d'illustrer ces quatre entités ainsi que leurs relations.

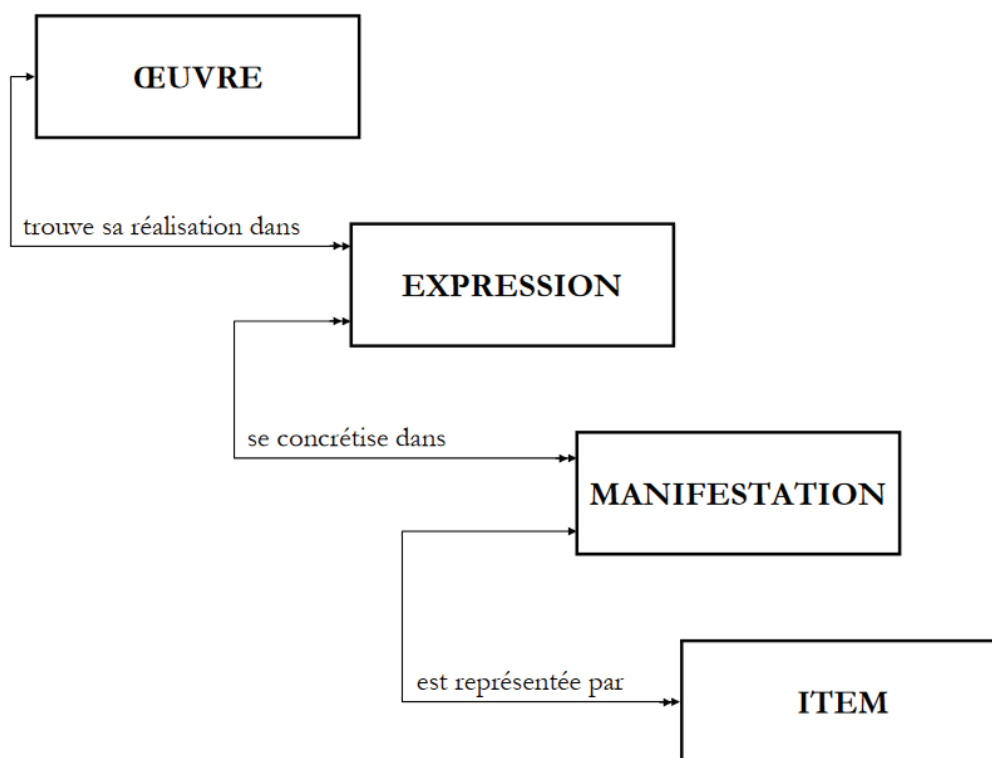


Fig. 8 - Exemple d'entité et de relation du modèle FRBR.
capturé depuis [Leresche et al., 2012, p.14]

Dans le schéma les flèches précisent le type de relation, soit *one-to-any*, soit *many-to-many*. Par exemple une *Oeuvre* peut avoir un ensemble d'*Expression*, mais des *Expressions* se rattachent à une seule et même œuvre⁵⁸.

On peut illustrer ces quatre entités et leurs relations avec l'exemple courant suivant : *Germinal* de Zola est une *Œuvre*; une traduction en anglais par tel traducteur est une *Expression*; la publication de cette traduction chez tel éditeur est une *Manifestation*; l'exemplaire que met à disposition telle bibliothèque est un *Item*.

On comprend alors un intérêt de ce modèle : il permet avec sa structuration de *relier* une autre traduction de l'ouvrage à l'*Oeuvre* de Zola, et ainsi de suite.

⁵⁸ Pour l'édition musicale par exemple on trouve dans [Leresche et al., 2012, p.20] l'exemple suivant : différentes interprétations d'une même *Oeuvre* forment différentes *Expressions*; les rééditions d'un même enregistrement d'une même interprétation – e.g. remastérisation – forment différentes *Manifestations*.

1.2 Analogie

1.2.1 Un modèle conceptuel

Le rapprochement le plus signifiant entre ce modèle bibliographique et le modèle que nous proposons se trouve dans la volonté d'apporter de l'ordre à un niveau conceptuel. En effet le modèle FRBR n'est pas l'introduction d'une nouvelle *norme* de description – ces normes étant parfois appelé *modèles de données* – mais la proposition d'un *cadre conceptuel* au sein duquel *penser et échanger* la réalisation de nouvelles normes de description par exemple :

Il s'agissait d'élaborer un cadre conceptuel permettant de comprendre clairement, sous une forme précisément exprimée et dans un langage qui soit parlant pour tout le monde *l'essence même* de ce sur quoi la notice bibliographique est censée renseigner. [Leresche *et al.*, 2012, p. 7] (nous soulignons).

Il ne s'agit que d'un cadre conceptuel visant à une certaine intercompréhension. Cela nous permet d'avoir une même structure à l'esprit et de nous référer aux mêmes concepts avec les mêmes noms. Cela nous permet de comparer des données qui peuvent ne pas être structurées de la même façon. [Le Boeuf, 2003, p. 4].

L'enjeu le plus essentiel pour un tel cadre est donc de s'établir à un niveau conceptuel. Le passage à ce niveau se retrouve essentiellement sur deux points. D'une part dans la volonté de faire face à l'hétérogénéité des types de supports ; donc de décrire le document au-delà de sa matérialité pour atteindre le contenu intellectuel ou artistique. Et d'autre part dans la volonté de dépasser les nombreuses normes de descriptions bibliographiques avec lesquelles traitent les catalogues de bibliothèque ; de dépasser la représentation que nous donnent ces normes⁵⁹. Dans cette lignée, la différence entre un modèle conceptuel et un modèle de donnée, n'est autre que ces derniers sont le fruit d'*implémentation*, destinés à la communication entre machines :

Peut-on qualifier les FRBR de « modèle de données » ? Il semble que les attributs qui y sont définis pour chaque entité soient dans bien des cas trop génériques pour permettre d'implémenter le modèle tel qu'il se présente, sans avoir à l'affiner davantage. [Le Boeuf, 2003, p. 4].

59 On retrouve dans la citation suivante la volonté de ne pas se baser sur les représentations que donnent les normes de descriptions existantes : « L'analyse s'attache en premier lieu non pas à des données particulières mais aux « choses » que décrivent les données » [Leresche *et al.*, 2012, p. 12].

Nous pouvons ainsi comprendre ce modèle comme une volonté de créer des concepts, une terminologie, qui permet de *penser* les descriptions de documents selon des besoins définis.

1.2.2 Schéma

Le schéma ci-dessous précise l'analogie entre le modèle FRBR et un modèle conceptuel pour les artefacts.

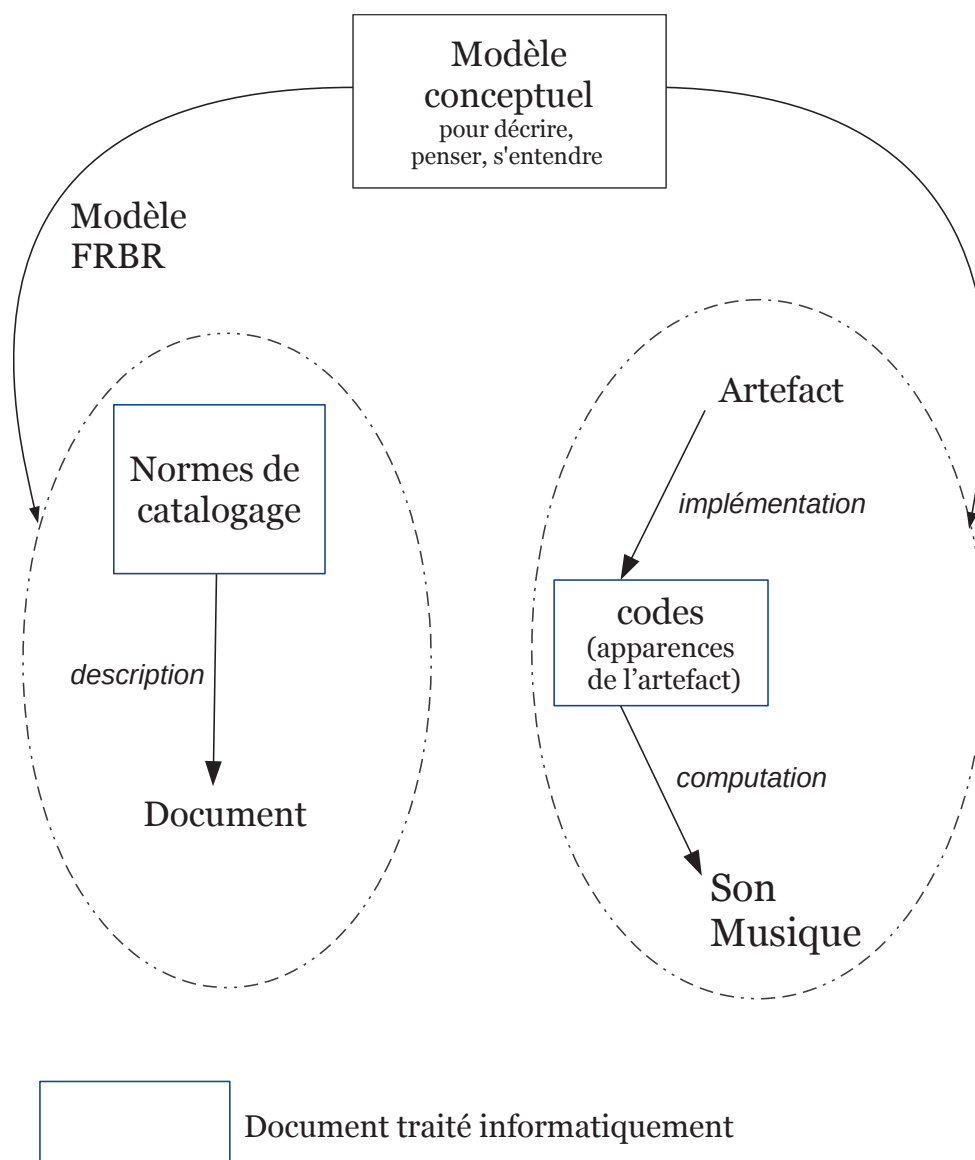


Fig. 9 - Analogie entre le modèle FRBR et un cadre conceptuel pour décrire les artefacts

La flèche légendée « Modèle FRBR » pointe sur un espace et non sur un document, afin d'indiquer que le modèle ne concerne ni précisément les documents ni précisément les normes de descriptions. Le modèle vise en effet un certain espace, constitué d'un ensemble de termes, de

concepts, qui permettent, pour une communauté, de *penser* les documents, et ceci non pas selon leurs inscriptions physiques, leurs matérialités, ni même selon les normes de description existantes, mais dans quelque chose de nettement plus abstrait, que l'on peut comprendre comme les connaissances véhiculées par les documents et avec lesquelles la communauté se structure.

2 Définition du cadre conceptuel

Le cadre que nous proposons se constitue de trois éléments clés : des Modules, des Algorithmes et le Routing. Les deux premiers sont les plus importants, car ils agrègent à eux deux l'essentiel des significations musicales. Le routing quant à lui, du fait de sa proximité avec le domaine de l'ingénierie, est le plus aisé à cerner.

2.1 Des Modules

Comme nous l'avons vu ce terme est déjà très fortement présent en informatique musicale du fait essentiellement de sa proximité avec les Ugen du langage historique Music N (cf. Chap.II 2.2). Nous présentons ci-après les similarités entre ces derniers et les Modules du cadre, puis nous décrivons différents types de Modules.

2.1.1 Définition

2.1.1.a Entre paramètre et signal audio

Les Modules de notre cadre croisent les Ugen en ce qu'ils sont en lien avec la production du signal audio. Nous pouvons ainsi les qualifier de modules audio. Ce point les rend particulièrement significatifs puisque ce sont eux qui sont responsables de la computation du signal audio, qui deviendra le son perçu. La perception du son passe donc nécessairement par l'usage de Modules.

Parallèlement, de l'autre côté de la boîte, en entrée (input), les modules possèdent un certain nombre de *paramètres*, qui permettent au compositeur « d'avoir la main » sur la production sonore.

2.1.1.b Système HAS

Au sein du système HAS les Modules se situent donc très proches du son. Une de leurs extrémités (coté son) se trouve juste en amont de la production des signaux audio de la pièce. L'autre extrémité (coté compositeur) est, elle, nettement plus floue. Les modules sont en effet équipés de *paramètres*, et non de contrôles : ils correspondent dans la composition à ce moment flou que nous avons mis en évidence (cf. Chap. III 3.2.3), où le matériau est travaillé, et où à celui-ci sous-tendent des jaillissements de significations, entre computation et perception. Pour souligner la proximité des paramètres au sonore, on peut les qualifier de *paramètres de synthèse*. Ils se situent en effet à un niveau où le son n'est pas encore absorbé par des stratégies de hauts niveaux.

2.1.2 Types de Modules

Tous les Modules ne sont pas de la même sorte. En leur appliquant une vue cybernétique on s'aperçoit en effet que tous n'ont pas les mêmes types d'entrées et de sorties. Il est ainsi d'usage de différencier des types de Modules. Par exemple, comme nous l'avons vu brièvement, dans [Schnell *et al.*, 2002] quatre fonctions de modules sont différenciées : synthèse, transformation, mémoire et analyse.

Dans notre contexte, où nous souhaitons apporter une terminologie pour l'appréhension des artefacts à des fins d'analyse musicale, nous différencions seulement deux types de Modules clés : les modules de *synthèse* et ceux de *traitement*⁶⁰.

2.1.2.a Synthèses

Les modules de synthèses produisent un signal audio en calculant les valeurs des échantillons qui constituent ce signal. Ils implémentent ainsi, *a minima*, une méthode de synthèse.

En entrée ils reçoivent des signaux de données et en sortie ils engendrent des signaux audio. Avec une vue informationnelle, comme le propose [Orlarey, 2009], on peut se représenter ces Modules comme des triangles, avec le sommet en haut, correspondant aux *paramètres* (discret) et la base en bas pour indiquer le signal audio (continu), auquel correspond en puissance la perception.

2.1.2.b Traitements

Les modules de traitements ont la spécificité d'avoir en entrée comme en sortie les mêmes types de signaux, des signaux audio. Ils traitent un signal déjà existant, celui qui est reçu en entrée. En plus de cette entrée audio ils reçoivent évidemment des paramètres qui permettent au compositeur d'avoir la main sur la transformation.

Comme ces Modules sont actifs sur un signal déjà existant, ils ont pour particularité que l'on ne peut savoir « comment ils sonnent ». En effet même si l'on connaît la méthode utilisée (*e.g.* delay) et les configurations des paramètres, le résultat, la sonorité, dépend inévitablement du signal entrant.

2.1.3 Vers des Modules conceptuels

La vue que nous proposons où l'artefact serait un assemblage de modules de synthèse et de traitement est déjà très courante en informatique musicale. Elle permet par exemple de structurer les connaissances nécessaires à la computation sonore (DSP). Dans la monographie

⁶⁰ De façon quelque peu anecdotique, on relève que dans [Jaffe, 1995] la distinction synthèse ~ traitement est utilisée pour différencier deux termes clés : *paramètre* et *contrôle*. L'auteur propose d'utiliser le terme *paramètre* quand il s'agit de synthèse, et *contrôle* pour les traitements.

de [Collins, 2010], *Introduction to Computer Music*, on retrouve ainsi un chapitre dédié à la synthèse et un au traitement. On pourrait donc voir le début de notre cadre comme une redondance, et même lacunaire puisque des types de modules restent absents, comme l'analyse de signal. Ce serait une fausse route pour la raison fondamentale que les Modules que nous proposons ont pour fin de dépasser ce niveau de computation. Il ne s'agit pas de rendre compte de fonctions mathématiques de signaux (analyse/synthèse/traitement), mais bien plus d'éclairer les moyens façonnés par un compositeur pour produire son matériau. La différence relève du niveau d'abstraction : les Modules du cadre sont d'un plus haut niveau d'abstraction que les modules de l'ingénierie audio⁶¹.

L'articulation paramètre/Module doit donc se faire dans l'optique de mettre à jour les éléments les plus saillants de l'artefact, et non les détails réalisés lors de l'implémentation. Ainsi les distinctions de fonctions ne sont pas forcément nécessaires, et un Module peut correspondre à un regroupement de fonctions. Ce regroupement que nous visons peut se comprendre par analogie avec le schéma de Music N : il ne s'agit pas d'effectuer des groupements au niveau des Ugen, mais bien plus au niveau des *instruments*.

Les règles qui régissent la délimitation et l'identification des Modules doivent donc s'approcher de « la composition », de l'acte compositionnel, et non chercher à être en cohérence avec un domaine computationnel, relatif au calcul des signaux audio.

Nous retrouvons cette distance avec le domaine de l'ingénierie, nécessaire pour atteindre les significations musicales, dans l'article de [Schnell *et al.*, 2002], portant sur le concept d'instrument composé :

After all it will be always the semantic one associates to a given stream and processing module which counts. *Ibid.*p. 4.

Deux exemples peuvent être donnés pour illustrer ce gain d'abstraction.

Un lecteur de sample, qui lit, ajuste un signal pré-enregistré, est considéré dans notre cadre comme un Module de synthèse sans que cela ne pose problème⁶².

Le dispositif de la pièce *Studie II* (1954) de K. Stockhausen consistant, pour l'essentiel, en un générateur de sinusoïdes et une réverbération (chambre d'écho de l'ordre de 10s.), est dans notre cadre considéré comme un seul Module de synthèse, puisque *la composition* s'est faite avec ce groupement. En effet, comme on peut l'écouter et le lire dans l'énoncé de la partition, la

61 Le lecteur relèvera la distinction avec le 'm' majuscule : Module du cadre ; module de l'ingénierie audio.

62 Au niveau de l'ingénierie cette association peut poser problème puisque les échantillons produits ne sont pas calculés (comme une synthèse) mais lus à partir d'une mémoire. (C'est ainsi que dans [Schnell *et al.*, 2002] la mémoire est une des quatre catégories de modules avancés. cf. Chap.II 2.2.3).

réverbération ne vient pas traiter, enrichir, un signal déjà diffusé, mais bien créer le matériau de la pièce, en laboratoire, qui sera ensuite développé dans le temps selon les indications de la partition⁶³.

2.1.4 Abrégé

- Au sein du système HAS, les Modules se situent entre l'Homme et le Son ;
- De deux types, synthèse ou traitement ;
- Implémentent des méthodes (synthèses et/ou traitements) ;
- Appartiennent au domaine des *paramètres* ;
- Intègrent des paramètres et engendrent des signaux audios ;
- La sémantique des paramètres vient de la méthode implémentée et de l'écoute du signal produit ;
- Ces Modules relèvent d'un niveau conceptuel. Ils se situent ainsi au-dessus de la définition stricte de synthèse et de traitement propre au domaine de l'ingénierie (DSP).

2.2 Des Algorithmes

2.2.1 Définition

Les algorithmes sont des chaînes de calcul fini qui répondent à un besoin. Ils sont divisibles en étapes, ou bloc d'opération, et l'algorithme est alors un processus qui englobe toutes ces étapes.

2.2.1.a Au-dessus des Modules

Comme le témoignait la citation de [Smith, 1991] dans le chapitre II (cf. Chap.II 2.3.2), il est fréquent dans la littérature de trouver l'expression « algorithme de synthèse sonore » (« sound generated algorithm »). En effet un Module de notre cadre peut être vu comme un algorithme de synthèse dans la mesure où il correspond à une suite de calcul fini (calcul des échantillons) et répond à un besoin (produire du son). Cependant, dans notre contexte, et dans la même mesure que les modules audio s'inscrivent au-delà de connaissances propres au DSP, nous excluons l'usage des algorithmes pour la production de signaux audio. Les algorithmes se trouvent ainsi

63 On retrouve ici une critique faite par [Sousa Dias, 2007] sur une « reconstruction » de *Studie II*. En effet, dans la reconstruction menée par Georg Hajdu, sur Max, ce ne sont pas les matériaux enregistrés en laboratoire avec la longue réverbération qui sont diffusés, mais des sinusoïdes réverbérées (cf. <http://georghajdu.de/6-2/studie-ii/>, 2017 07 01).

« au-dessus » des Modules. Ils concernent la production des paramètres de synthèse (signaux de données) et sont ainsi d'un niveau d'abstraction plus élevés.

2.2.1.b Système HAS

Dans le système HAS les algorithmes se trouvent alors au plus proche de l'Homme, du compositeur. Ils avoisinent les idées compositionnelles, ou encore les stratégies de haut niveau. Nous pouvons les voir comme formant une couche additionnelle, construite par le compositeur, qui viendrait s'apposer sur les paramètres de synthèse des Modules. Les algorithmes intègrent donc des *contrôles*, sur lesquels le compositeur va pouvoir agir.

À la différence des Modules, les Algorithmes n'implémentent pas de méthodes qui appartiendraient à un domaine de connaissance particulier (e.g. DSP). Ils sont *de facto* fortement singuliers aux compositeurs ; ils possèdent le plus d'idiosyncrasie. Par la-même, ils peuvent être des plus significatifs pour la composition.

2.2.1.c Des processus temporels

Une particularité des algorithmes qui permettra de rendre compte de leurs hauts niveaux d'abstraction, de leurs distances avec la production audio, se trouve dans le fait qu'ils peuvent, non systématiquement cependant, engendrer des structures musicales de l'ordre de la phrase (*i.e.* de l'ordre de 2s.). Schématiquement, si les Modules audio concernent la production sonore de l'ordre de la microstructure, les algorithmes peuvent eux concerner la mesostructure, sinon la macrostructure.

Cependant rien n'est aussi limpide dans la densité des artefacts. On pourra aussi trouver des algorithmes légers en densité, qui permettent d'effectuer des interrelations entre paramètres, tout en restant en deçà de stratégie de haut niveaux.

2.2.2 Abrégé

- Au sein du système HAS les Algorithmes se situent entre l'Homme et les Modules ;
- Appartiennent au domaine du *contrôle*, loin du signal audio ;
- Sont connectés aux nombreux *paramètres* des Modules ;
- La sémantique des contrôles se situe à la croisée de (i) ce que réalise l'algorithme, (ii) des connexions établies avec les paramètres des Modules, et (iii) de l'écoute du signal produit ;
- Possèdent un fort taux d'idiosyncrasies.

2.3 Le routing

Le dernier élément clé du cadre conceptuel concerne les connexions entre Modules. Nous souhaitons utiliser le terme *routing* pour désigner ces connexions, et plus largement l'acheminement du signal audio vers un système de diffusion. Ce terme est emprunté à l'ingénierie audio. Il est fortement présent dans les stations audionumériques de travail (ou Digital Audio Workstation, DAW par la suite) et par conséquent il est familier de quiconque travaillant dans la technologie musicale. C'est par cette familiarité que le *routing* est certainement l'élément le plus simple de notre cadre.

2.3.1 Définition

Dans un DAW le routing concerne deux possibilités. Il s'agit d'abord des connexions entre pistes, par exemple la liaison d'une piste audio vers une piste effet, ou encore la sommation de pistes audio dans un bus commun. Il s'agit ensuite des connexions entre pistes audio et sorties physiques du dispositif. Ces dernières correspondent aux nombres de canaux audio nécessités par l'artefact, typiquement deux pour une diffusion stéréophonique.

De la même manière le *routing* de notre cadre conceptuel se divise en deux. Il y a d'abord l'ensemble des connexions entre Modules, le *routing interne*, et ensuite l'ensemble des connexions entretenues avec le système de diffusion, ou spatialisation, le *routing externe*.

2.3.1.a Routing interne

Dès lors qu'il y a des Modules qui peuvent modifier un son pré-existant — les Modules de *traitements* – il est possible de penser la production du son selon des interconnexions entre Modules, à l'instar d'un réseau. Cette flexibilité du *routing interne*, si elle peut être difficile à imaginer en musique, est par contre inévitable dans les jeux vidéo, comme en témoigne Andy Farnell :

Most game audio systems incorporate a mixer much like a traditional large frame multi-bus desk with groups, auxilliary sends, inserts and busses. The difference between a digital desk used for music and one used in game audio is more to do with how it is used. In traditional music the configuration of the desk stays largely the same throughout the mix of a piece of media, but in a game the entire structure can be quickly and radically changed in a very dynamic way. Reconfiguring the routing of the entire mix system at the millisecond or sample accurate level without clicking or dropouts is the strength of game audio mixers [Farnell, 2007, p. 18].

Le routing des pièces musicales traditionnelles – pour rendre les mots de l’auteur – apparaît donc comme figé comparé à celui du jeu vidéo, puisque les reconfigurations des connexions entre Modules restent rares, et *a fortiori* non prises en compte dans la composition.

Cette distinction entre un routing interne fixe et un dynamique, vient témoigner d’une des particularités de la musique d’informatique, où un routing dynamique est tout à fait envisageable. Avec l’outil informatique il est en effet possible de penser la production sonore avec un routing configurable au cours du temps. C’est par exemple le cas dans *Jupiter* où un réseau de Modules est implémenté de sorte que les connexions entre Modules peuvent être changées dans l’écoulement de la pièce.

Cette possibilité de configurer à loisir le *routing interne* est aussi pointée dans l’article de James McCartney présentant le langage SuperCollider [McCartney, 2002, p. 62-63]. L’auteur donne un exemple où il est possible de paramétrer le nombre de filtre en peigne et le nombre de filtre passe tout à appliquer en série sur le même signal source, et ceci simplement à l’aide de deux variables^{64 65}.

2.3.1.b Routing externe

Le *routing externe* amène à considérer un aspect que nous avons volontairement laissé de côté jusqu’à présent, celui de la spatialisation, de la diffusion des signaux. Si l’on considère la spatialisation l’artefact n’est plus quelque chose qui produit un *signal*, mais quelque chose qui projette des sons dans un espace, *via* un dispositif. Dès lors, différentes questions apparaissent : Comment les haut-parleurs sont-ils disposés dans l’espace ? Comment l’espace est-il investi ? Avec combien de dimensions et avec quelles technologies ?

Pour la réalisation d’*Inharmonique* (1977) la question de la spatialisation se pose peu puisque la pièce est stéréophonique. Mais pour *Jupiter* (1987), par exemple, la classique stéréophonie est augmentée d’un système quadraphonique. Et de nos jours, trente ans après, on sait que la composition peut aussi se faire sur la spatialisation (cf. le dernier chapitre de [Solomos, 2013] par exemple) ; au panoramique s’est substitué le *spatialisateur*, qui permet au compositeur de placer ces sons dans un espace quadri, octo, n-phonique.

Le *routing externe* nous amène donc à la limite de notre démarche théorique. Le cadre conceptuel que nous proposons ne tient pas compte de la spatialisation. Il pourra ainsi être revu dans une démarche de plus grande ampleur. Face à notre problématique, déjà conséquente, nous avons en effet fait le choix, pour commencer, pour jeter les premières briques

64 Le routing n’est ici pas totalement dynamique puisque l’ordre des traitements reste figé ; seul le nombre de module à insérer est paramétrable.

65 Ce qui ne s’avérerait pas possible dans les langages comme Csound ou Pure Data.

conceptuelles, de ne pas tenir compte des questions posées par la spatialisation et de ne considérer qu'une diffusion standard, stéréophonique.

2.3.2 Abrégé

- Le *routing interne* concerne les interconnexions entre Modules ;
- Le *routing externe* concerne les liaisons des Modules aux haut-parleurs ;
- En musique d'informatique le *routing interne* peut être partie intégrante de la composition.

2.4 Processus dynamique, imbrication

La définition du cadre doit maintenant aborder les relations possibles entre les Modules et les Algorithmes. En effet l'état actuel de notre cadre est problématique puisque ces relations entre Modules et Algorithmes apparaissent comme figées : les Algorithmes se situent « au-dessus » des Modules. Or, comme nous l'avons montré dans le chapitre précédent, le travail compositionnel est sous-tendu par un processus hautement dynamique, qui a lieu au sein du système HAS. Du côté de l'analyse, le problème d'un cadre qui serait figé se manifeste par le fait qu'il n'est pas possible dans un premier temps de retrouver cette différence de niveau entre Algorithmes et Modules. De prime abord l'analyste doit passer par une intégration des significations des *variables*, qui parcourent la densité de l'artefact, ce qui appartient à un niveau d'abstraction nettement plus bas que celui où *prend sens* la distinction entre Algorithmes et Modules.

Au sein du système HAS, et de la même façon que la composition se fait avec un processus fortement dynamique – c'est-à-dire, avec la théorie de Peirce, qu'il y a jaillissement de *representamen-interpreting* – il faut que l'analyse se fasse aussi avec un tel processus, entre Son et Artefact, afin que des niveaux d'abstraction puissent être discernés. Dans notre cadre il est alors fondamental de ne pas figer la relation entre Algorithmes et Modules : les Modules peuvent imbriquer des Algorithmes et ainsi de suite autant que nécessaire.

De même qu'il y a, au niveau des paramètres, imbrication de significations durant le travail sur la compréhension des Modules – comme on a pu le voir avec l'artefact d'*Inharmonique* (cf. Chap.II 4.1) – il est nécessaire de poursuivre cette imbrication de significations avec les Algorithmes. C'est en autorisant ce procédé d'imbrication de significations que les *contrôles*, les éléments saillants de l'artefact, pourront être éclairés.

Ce besoin de dynamique dans l'analyse se retrouve corrélativement par le fait que l'artefact ne peut se lire linéairement. Relevons :

- l'artefact n'est effectif qu'au travers d'une implémentation qui déterminera son apparence ;
- son apparence est le fruit d'un travail de *design* ;
- la structure de cette apparence ne reflète pas nécessairement la structuration de l'artefact, c'est-à-dire ses niveaux d'abstraction.

2.5 Représentation

Nous pouvons à présent représenter le cadre au sein du système HAS.

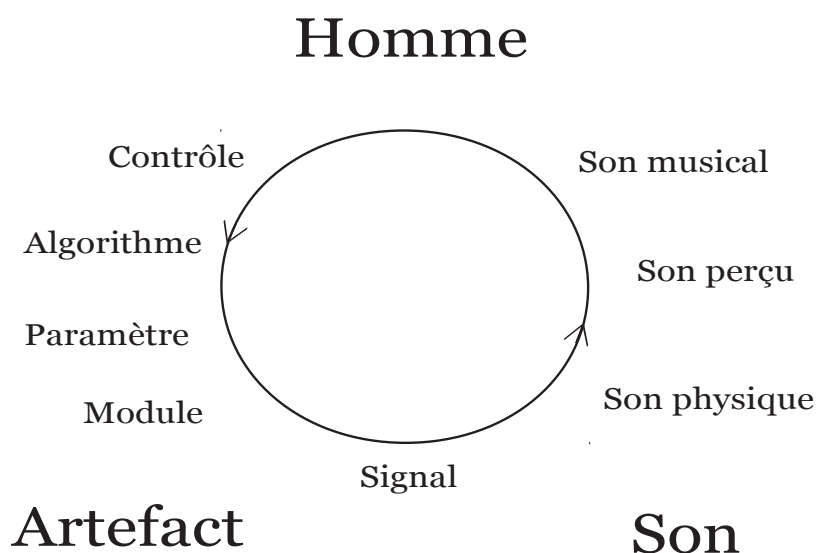


Fig. 10 - Représentation du cadre conceptuel au sein du système HAS

Cette représentation rend compte de l'articulation des deux éléments clés du cadre, Algorithme et Module. Comme nous l'avons vu, les Algorithmes, liés aux *contrôles*, sont d'un niveau d'abstraction plus élevé que les Modules, liés aux *paramètres*. Dans l'autre relation du système, entre l'Homme et le Son, nous avons placé les différentes considérations du Chapitre II sur le matériau. Le *son physique* est le résultat de la transduction du *signal* calculé par les Modules. C'est un son que l'on considère par ses propriétés physiques. Le *son perçu* à la différence est lui considéré selon son écoute. Enfin le *son musical* est un son, ou plutôt un ensemble de sons, avec lequel les idées compositionnelles rentrent en cohésion. Cet ensemble de son résulte du travail compositionnel, de l'action du compositeur.

Une erreur de lecture serait de voir ces différents niveaux arrêtés, et de penser que le *son physique* correspond au Module et ainsi de suite. Ce serait oublier l'aspect essentiel du cadre, sa

dynamique. *Paramètres* et *Modules* s’imbriquent, autant que nécessaire, et de même entre *Module* et *Algorithme*. Cet aspect dynamique est représenté par les deux flèches sur le cercle.

3 Un concept pour éclairer l'aspect dynamique

3.1 Introduction

Le cadre que nous avons amené possède un aspect à la fois central et difficile à cerner : son aspect dynamique. Il est central puisque ce n'est que en acceptant une profusion de significations que la distinction structurante entre Algorithme et Module peut être faite, c'est en effet avec ce processus hautement dynamique que les niveaux d'abstraction pourront être distingués. En outre, cet aspect reste difficile à cerner puisqu'il n'est pas évident de saisir théoriquement ce qui peut gouverner ce « gain d'abstraction » nécessaire. Si nous avons vu ce qu'il ne fallait pas utiliser pour discerner les Modules – la différence entre synthèse/traitement/analyse en ingénierie audio – nous n'avons cependant pas vu ce qu'il fallait utiliser.

Ce grand titre se propose d'avancer dans cette direction. Il ne s'agit pas cependant de répondre de façon formelle. En effet il apparaît que la question « que faut-il utiliser pour discerner les Modules entre eux, puis les Modules des Algorithmes ? » ne peut avoir de réponse évidente. Ce n'est peut-être que dans la pratique, lorsque l'analyste a déjà intégré une quantité conséquente de significations – soit lorsque la question se pose de l'intérieur – qu'une réponse peut être donnée. Plutôt qu'une réponse, nous proposons donc ici une voie à emprunter pour avancer dans ce gain d'abstraction.

Pour ce faire nous utilisons un couple de concepts qui était présent dès l'origine de nos travaux, celui de *potentiel* et d'*actualisation*. Dans un premier temps nous le présentons *via* Marco Stroppa ; dans un deuxième nous donnons notre propre exposition ; dans un troisième nous effectuons des liens avec d'autres couples de concepts présents dans l'étude du langage – chez F. de Saussure et N. Chomsky –, et finalement nous utilisons ces concepts pour éclairer cet aspect dynamique.

3.2 Potentiel et instance chez M. Stroppa

Nous avons trouvé les bases de ce couple de concepts dans une interview du compositeur M. Stroppa sur la Composition Assistée par Ordinateur, [Stroppa, 1999]. Comme il est question d'une interview les concepts ne sont pas rigoureusement définis, mais bien plus présentés dans le fil de la discussion. Pour illustrer ce couple de concepts le compositeur utilise le terme *instance* plutôt que *actualisation*. Néanmoins, comme nous le verrons, l'idée du concept reste similaire.

La présentation des concepts est précédée d'une situation imaginée que nous pouvons ainsi résumer – le texte entre guillemets provient du compositeur :

« Supposons qu'un compositeur, avec une bonne maîtrise technique mais peu d'expérience concrète, veuille réaliser une œuvre utilisant la synthèse ». Afin d'obtenir un son approprié il va d'abord essayer « de trouver la ou les techniques de synthèse les mieux adaptées à son idée originale ». Ensuite, après un premier résultat sonore, il va créer de nouveaux sons « dérivés du son initial », de sorte à établir différents liens entre eux.

C'est dans l'optique de favoriser cette *mise en liens* entre des sons que M. Stroppa amène le concept de *potentiel sonore*. L'enjeu est de « penser différemment » les sons, de les envisager non pas comme des sons isolés, mais, à l'inverse, comme des *instances* d'une *abstraction unique*. Le *potentiel sonore* n'est alors autre chose que cette abstraction unique, de laquelle découlent les nombreuses instances :

J'appelle cette abstraction donnant lieu à un ensemble de sons un *potentiel sonore*. C'est un concept abstrait qui saisit l'idée d'une famille de sons constituant un développement dans une direction précise. *Ibid.* (souligné dans le texte).

Les relations entre le *potentiel* et ses *instances* sont telles qu'il existe un rapport de contiguïté entre les différentes instances : elles forment une « famille de sons », elles partagent des « caractéristiques morphologiques » communes, de sorte à assurer « une permanence perceptuelle ». (p. 67).

Le potentiel sonore se laisse ainsi assimiler facilement aux instruments traditionnels, qui assurent une certaine cohérence dans le matériau musical (p. 67). Cependant, le compositeur soulève une différence importante : « si l'instrument est un concept assez neutre du point de vue esthétique, un potentiel sonore contient déjà *in nuce* la trace d'une poétique, sinon d'une esthétique » (p. 67). Dans la même lignée, il souligne que le « potentiel ne peut se concevoir sans avoir une idée du projet compositionnel ».

3.3 Potentiel et Actualisation, exposition

3.3.1 Potentiel sonore

Un *potentiel sonore* est quelque chose d'abstrait, qui se forme lors de la réalisation d'une composition. Il est une façon de voir l'engendrement des sons par voie computationnelle. Une sorte de *réservoir* sonore, présent en puissance, dans la cognition du compositeur, qu'il utilise à loisir pour faire des sons.

M. Stroppa marque le potentiel comme fortement issu de l'écoute des sons. À cette vue perceptuelle nous ajoutons, suite au système HAS, une considération sur l'artefact : le potentiel se constitue au sein du système HAS. C'est avec les jaillissements de significations issus d'une part de l'artefact et d'autre part de la perception, rentrant en interaction avec une idée compositionnelle, que le potentiel sonore se forme.

3.3.1.a Potentiel et méthode de synthèse

Le potentiel sonore sous-tend au minimum l'implémentation d'une méthode de synthèse. Cependant, il ne peut en aucun cas s'y réduire. Il est par exemple aisé d'envisager un réservoir de sons possibles après un apprentissage de la méthode de synthèse par modulation de fréquence (FM). En effet, dans cette synthèse, le timbre des sons résultants est caractéristique. Cependant, lorsqu'un compositeur utilise cette méthode de synthèse, il l'implémente, il la réalise, la confronte à une idée compositionnelle, en ajoutant un ensemble d'éléments⁶⁶ qui lui est propre, il effectue un travail de *design*.

Une méthode de synthèse n'appelle donc pas un réservoir de sons en puissance, mais c'est l'utilisation d'une méthode de synthèse par un compositeur, intégrée dans un projet compositionnel, qui rend une idée compositionnelle effective, qui appelle un réservoir de sons, un *potentiel sonore*.

Cette différence est présente chez M. Stroppa quand il est dit qu'un potentiel contient « déjà *in nuce* la trace d'une poétique, sinon d'une esthétique ». Le potentiel n'est en effet pas quelque chose de neutre, que le compositeur intègre, et qu'il serait possible de se passer entre compositeurs. À l'inverse nous dirons que le potentiel est *habité* par un compositeur, il est présent en puissance dans la cognition, lors d'une activité de création.

3.3.2 Actualisation

À l'inverse du potentiel l'actualisation n'est pas quelque chose d'abstrait mais concerne directement la *perception*, l'écoute du son. Elle s'adresse donc à nos sens. L'idée d'actualisation découle en fait de celle du potentiel : pour que le potentiel puisse être du son en puissance, il faut néanmoins que ces sons puissent être tôt ou tard écoutés. C'est alors la présentation des sons à notre écoute que l'on nomme *actualisation*⁶⁷.

Entre le *potentiel* et les *actualisations* il réside une relation de *one-to-many* : les sons engendrés ne sont pas isolés mais possèdent des similarités morphologiques, de sorte à former

66 Par ex. second oscillateur modulant, ou nouvelle enveloppe d'amplitude, corrélation entre paramètres, algorithme d'engendrement de valeurs de fréquence etc.

67 Nous préférons le terme *actualisation* à celui d'*instance*, car le second se rattache, pour nous, trop facilement au paradigme de la programmation orientée objet. Or le concept de *potentiel* dépasse toute technique de programmation.

une « famille de sons ». Si M. Stroppa souligne bien cette « permanence perceptuelle », nous souhaitons insister sur la similarité du processus d'engendrement. En plus de partager des similarités morphologiques, les sons d'un même potentiel partagent le même processus d'engendrement, la même technique. On peut ainsi passer de l'un à l'autre avec une certaine continuité dans la « paramétrisation ».

La vue proposée par [Depalle et Rodet, 1993] et reprise par [Bresson, 2007] est ici éclairante. Au niveau de l'implémentation – en comprenant qu'un son correspond à l'attribution de valeurs à un ensemble de paramètres⁶⁸ – on peut envisager que la production d'un autre son se fait avec l'attribution d'autres valeurs à ces mêmes paramètres, et ainsi de suite.

Les rapports de contiguïté entre les différentes actualisations sont donc présentes et dans la perception et dans l'action sur l'artefact.

3.3.3 Potentiel sonore et actualisation dans le cadre conceptuel

Nous finalisons l'exposé de ces deux concepts en les situant dans notre cadre. Nous voyons le *potentiel* comme appartenant au niveau des *paramètres*, et non pas à des hauts niveaux d'abstraction véhiculés par les algorithmes. Le *potentiel* est ainsi attaché au *sonore*, et non pas à des structures de haut niveau.

Le couple de concepts est alors utile pour investir un Module que l'analyste doit éclairer. L'analyste est confronté à quelque chose qui, et c'est fondamental, au sein d'une pratique compositionnelle, génère du sonore et il doit chercher à éclairer un *potentiel*, un réservoir de sons que le compositeur s'est dessiné.

Avec cette compréhension du *potentiel* nous pouvons comprendre l'Artefact comme un ensemble de *potentiels sonores*. L'intérêt de cibler des niveaux d'abstraction de plus haut niveau est qu'il est nécessaire par la suite de rendre compte de l'usage de ces *potentiels* entre eux, comme nous le verrons par la suite.

3.4 Des concepts du langage

Il nous a semblé très tôt que ce couple de concepts, entre quelque chose d'abstrait et quelque chose de perceptible, pouvait s'enrichir avec des concepts présents dans l'étude du langage. Nous avons en effet trouvé des ressemblances avec les concepts de F. de Saussure entre Langue et Parole, puis inévitablement avec ceux de N. Chomsky qui en ont découlé, entre Compétence et Performance. Ci-après nous présentons sommairement ces deux couples de concepts puis nous expliquons les parentés entretenues avec le couple potentiel/actualisation.

68 Des paramètres représentent le son (cf. Chap.III 2.1).

3.4.1 Langue et Parole chez F. de Saussure

Les concepts de Langue et Parole sont présents dans les premières pages du célèbre *Cours de Linguistique Générale* de F. de Saussure, [Saussure, 1969], sur lequel notre présentation se base exclusivement – les prochaines citations sont seulement suivies du numéro de page.

3.4.1.a Un concept pour investir le langage

La distinction entre *Langue* et *Parole* est une solution fondée par l'auteur pour faire face à un « amas confus de choses hétéroclites sans lien entre elles » (p. 24), qui survient lorsque le langage est étudié. Le langage possède en effet des spécificités qui rendent son étude difficile. Quatre points sont soulignés par l'auteur : le langage (i) se situe entre « impressions acoustiques » (réception) et « articulations buccales » (production) ; (ii) fait appel au son – une « unité complexe acoustico-vocale » qui est aussi bien physiologique que mentale ; (iii) a un côté social comme un côté individuel ; et enfin (iv) est à la fois une « institution actuelle » et un « produit du passé ».

C'est devant ces nombreuses facettes présentes dans l'étude du langage et dans une volonté de mettre de l'ordre que le concept de *langue* est d'abord introduit :

il faut se placer de prime abord sur le terrain de la langue et la prendre pour norme de toutes les autres manifestations du langage. (p. 25, souligné dans l'ouvrage)

dès que nous donnons à la langue la première place parmi les faits de langage, nous introduisons un ordre naturel dans un ensemble qui ne se prête à aucune autre classification. (p. 25).

À ce concept de *langue* vient ensuite s'ajouter celui de *parole*, avec une hiérarchisation : la *parole* est subordonnée à la *langue*. Et c'est ni plus ni moins avec ces deux concepts, et surtout avec leurs relations, que la linguistique Saussurienne se fonde :

En accordant à la science de la langue sa vraie place dans l'ensemble de l'étude du langage, nous avons du même coup situé la linguistique tout entière. Tous les autres éléments du langage, qui constituent la parole, viennent d'eux-mêmes se subordonner à cette première science *Ibid.* p36.

Le tout global du langage est inconnaissable, parce qu'il n'est pas homogène, tandis que la distinction et la subordination proposées éclairent tout. Telle est

la première bifurcation qu'on rencontre dès qu'on cherche à faire de la théorie du langage *Ibid* p. 38

C'est donc avec ce couple de concepts que Saussure vient faire face au problème de l'étude du langage, à ce tout hétéroclite.

3.4.1.b Définition

Afin d'amener ces deux concepts l'auteur présente d'abord un schéma de communication. Ce dernier, volontairement simplifié par l'auteur, divise la communication orale entre deux personnes en quatre étapes successives. Chez l'orateur d'abord, il y a (i) association de *concepts* à des *images acoustiques* (processus psychiques), puis (ii) transmission d'une impulsion, corrélative à l'image acoustique, du cerveau aux organes phonatoires (processus physiologiques). Entre l'orateur et le récepteur ensuite, il y a (iii) émission/réception des ondes sonores (processus physiques). Chez le récepteur enfin, il y a (iv) transmission physiologique de l'image acoustique via des impulsions, puis (v) association de cette image acoustique au concept.

La *Langue* se trouve alors dans la partie psychique de la communication, dans la première et la dernière étape. Elle concerne les multiples associations entre les concepts (ou « faits de consciences »), et d'autre part les images acoustiques (ou « représentations des signes linguistiques ») – ce qui deviendra par la suite de l'ouvrage les célèbres *signifié* et *signifiant*. La *Langue* concerne donc les nombreuses associations qui se tissent entre les deux faces du signe linguistique. Elle est un vaste *système de signes* qui a lieu au « niveau psychique » :

La langue est un système de signes où il n'y a d'essentiel que l'union du sens et de l'image acoustique, et où les deux parties du signe sont également psychiques. *Ibid.* p32

La *Parole* concerne elle l'ensemble des processus présents chez l'orateur entre l'étape du concept à la production de l'onde sonore, de l'étape une à la trois. Au niveau psychique, la Parole concerne le « coté exécutif », le passage des concepts aux images acoustiques.

Ajoutons pour finir que ces deux concepts se différencient pleinement avec l'opposition sociale ~ individu.

la langue n'est pas une fonction du sujet parlant, elle est le produit que l'individu enregistre passivement [...] la parole est au contraire un acte individuel de volonté et d'intelligence *Ibid.* p. 30

La *Langue* est partagée entre Hommes, elle relève du social tandis que la *Parole* concerne l'individu.

3.4.1.c Articulation

Un aspect essentiel de ce couple de concepts se trouve dans leur articulation. Il serait faux de croire que la Langue et la Parole s'acquièrent, ou se développent, indépendamment. En effet, tout à l'inverse, l'un s'enrichit de l'autre, et réciproquement.

La langue est nécessaire pour que la parole soit intelligible et produise tous ses effets ; mais celle-ci est nécessaire pour que la langue s'établisse ; historiquement, le fait de parole précède toujours. [...]

[la langue] n'arrive à se déposer dans notre cerveau qu'à la suite d'innombrables expériences. Enfin, c'est la parole qui fait évoluer la langue : ce sont les impressions reçues en entendant les autres qui modifient nos habitudes linguistiques. Il y a donc interdépendance de la langue et de la parole ; celle-là est à la fois l'instrument et le produit de celle-ci. p. 38

Entre la Langue et la Parole il y a donc interdépendance, et ceci doit être pris en compte dans ce que permet d'étudier ce couple de concepts, ici le langage. C'est cette interdépendance qui va nous permettre d'éclairer l'aspect dynamique du cadre.

3.4.2 Compétence et Performance chez N. Chomsky

Le second couple de concepts est amené par N. Chomsky dans les années 1960. Il est en fait affilié aux concepts de Saussure, mais s'en distingue aussi fondamentalement dans la mesure où la conception même du langage est différente chez ces deux chercheurs.

Avant de présenter un aspect de ces concepts il est nécessaire de prendre des précautions. D'une part, ce couple de concepts a irrigué dans la seconde moitié du XXe une importante partie de la cartographie de la science (au moins psychologie, science du travail, science de l'éducation, intelligence artificielle), avec des significations parfois différentes, et d'autre part, le sens de ces concepts a pu être modifié par l'auteur même⁶⁹. Conséquemment, la présentation que nous en faisons est lacunaire vis-à-vis de la réflexion de l'auteur ; il ne s'agit pas de témoigner de la place de ces concepts chez l'auteur, mais de les exposer au travers de trois ouvrages, [Chomsky, 1964], [Chomsky, 1965] et [Chomsky, 1967].

69 Une vingtaine d'années après leurs apparitions ces concepts ont par exemple été revus avec une nouvelle terminologie, I-Language et E-Language (cf. N. Chomsky, *Knowledge of language: Its nature, origin and use*, New York: Praeger, 1986. cité dans [Clark, 2007]).

Nous proposons d'éclairer ces concepts en deux temps. Le premier à partir d'une vue linguistique, entre grammaire universelle et grammaire particulière, et le second à partir de l'aspect créatif du langage, essentiel chez l'auteur.

3.4.2.a Des grammaires

La théorie linguistique de Chomsky se dessine avec une distinction entre deux grammaires, une première, dite grammaire *particulière*, et une seconde dite *universelle*. La première est celle habituelle qui s'étend depuis des siècles, elle permet de décrire les phrases qu'une langue⁷⁰ peut produire. La seconde est spécifique à l'auteur⁷¹, et on peut comprendre les trois précédentes références comme des textes visant à la fonder. La grammaire *universelle* est plus générale, elle dépasse la diversité des langues pour mettre en lumière les mécanismes communs qui apparaissent dans les descriptions des langues que donne la *grammaire particulière*. La *grammaire universelle* se situe donc à un niveau au-dessus de celle *particulière*, elle l'englobe⁷². La réflexion de l'auteur passe ensuite par une critique des grammaires particulières :

[Celles-ci] sont défectueuses : elles manquent à formuler bien des régularités fondamentales de la langue dont elles s'occupent. Cela est particulièrement clair au niveau de la syntaxe, où aucune grammaire, qu'elle soit traditionnelle ou structuraliste, ne parvient, par-delà la classification des exemples particuliers, à formuler des règles génératives de quelque extension. [Chomsky, 1965, p. 15].

Pour l'auteur les grammaires attachées à une langue s'avèrent lacunaires puisqu'elles passent à côté des régularités fondamentales d'une langue. Il résiderait après l'étude d'une langue, des éléments plus abstraits, non pris en compte par la grammaire particulière, et pourtant bien inhérents à la langue.

Nous pouvons éclairer cette lacune en essayant de démanteler cette notion riche de grammaire. En prémisses, comme nous l'amène l'auteur [Chomsky, 1957, p. 13], il faut voir que l'étude d'une langue peut se faire à différents niveaux, comme celle de la phonologie, de la morphologie ou encore de la syntagmatique. À ceux-ci nous pouvons en ajouter deux autres – sans chercher à savoir si après cet ajout les niveaux se recouvrent ou non –, présents dans [Chomsky, 1965, p. 31], la syntaxe et la sémantique. On peut alors voir la grammaire comme un prisme, construit avec certains de ces niveaux, et la lumière qui se présente au prisme comme la

70 Le terme langue est utilisé dans un sens conventionnel, en dehors des concepts de Saussure, dans le sens où le français, le chinois ou l'arabe sont des langues.

71 Ou du moins elle fût remise d'actualité au XXe par N. Chomsky. En effet l'auteur montre que l'idée d'une grammaire universelle était déjà présente au XVIIIe (cf. [Chomsky, 1965, p. 16]).

72 « une théorie linguistique donnée détermine à l'avance pour toutes les grammaires la manière dont chaque grammaire est reliée au corpus des phrases de la langue qu'elle décrit » [Chomsky, 1957, p. 16].

langue à étudier. Alors, les décompositions spectrales formées par le prisme seraient des représentations de la langue, obtenues par les différents niveaux imbriqués dans le prisme. La grammaire permet ainsi de donner différentes représentations d'une langue, et à chacune de ces représentations correspond un niveau linguistique. Un niveau est alors à la fois « un mécanisme valable pour la construction de grammaires » et, réciproquement, « une certaine méthode de représentation des énoncés » [Chomsky, 1964, p. 13].

Nous pouvons alors comprendre, dans un premier temps, la critique faite aux grammaires traditionnelles comme manquant d'un certain niveau de représentation. Ce niveau doit dépasser la diversité des langues de sorte à viser leurs régularités, et ceci en cohérence avec d'autres représentations grammaticales. La grammaire *universelle* est faite de trois composants, un premier, fondamental, syntaxique, et deux autres, plus courants, phonologique et sémantique. C'est le premier de ces composants qui est surtout nouveau pour la grammaire. Il est fondamental en ce qu'il assure la cohésion des deux autres représentations, phonologique et sémantique. Ce composant est abstrait, il est comme détaché du sonore des mots. Il « caractérise un ensemble infini d'objets formels abstraits, dont chacun contient toute l'information nécessaire à l'interprétation unique d'une phrase particulière ». On peut voir ce composant comme quelque chose qui engendre une certaine structure, qui sera reliée, *via* les deux autres composants, aux représentations phonologique et sémantique. (cf. [Chomsky, 1965, p. 30] pour une explication de ces composants et de leurs articulations).

3.4.2.b L'aspect créatif du langage

Le second aspect clé de cette linguistique concerne ce qui est appelé *la nature générale du langage*⁷³. Cet aspect est fondamental puisqu'il vient aussi expliquer le besoin d'une grammaire *universelle*. Il nous a cependant semblé opportun d'abord d'amener la distinction entre une grammaire universelle et particulière, et ensuite de comprendre la première avec la conception du langage de l'auteur.

En liminaire des deux ouvrages [Chomsky, 1964] et [Chomsky, 1965], l'auteur part du schéma suivant de communication orale.

A mature speaker can produce a new sentence of his language on the appropriate occasion, and other speaker can understand it immediately, though it is equally new to them. [Chomsky, 1964, p. 1].

Dans ce schéma standard, ce qui attire particulièrement l'attention de l'auteur c'est le fait que le locuteur puisse produire une infinité de phrases ou, réciproquement, que l'auditeur puisse

73 « Nous ne nous intéressons pas seulement à des langues particulières, mais aussi à la nature générale du langage » [Chomsky, 1957, p. 16].

comprendre une infinité de phrases, alors que le système cognitif du locuteur-auditeur, sa mémoire par exemple, sont eux finis. Ce paradoxe de la langue est résumé avec l'expression de W. de Humboldt : une langue « fait un usage infini de moyens finis » (e.g. [Chomsky, 1964, p. 9]). Ce fait, s'il peut paraître au premier regard relativement commun, s'avère central dans la théorie linguistique de Chomsky. On le retrouve ainsi de façon récurrente dans les écrits susmentionnés⁷⁴. Il est ensuite nommé par l'auteur « l'aspect *créatif* du langage ». Un locuteur-auditeur donné qui maîtrise une langue a la possibilité de produire une infinité de phrases pourvues de la sémantique souhaitée.

La centralité de cet aspect créatif se retrouve au moins sur deux points. D'une part c'est avec cette conception créative du langage que Chomsky critique la théorie de Saussure⁷⁵, et de l'autre c'est parce que la grammaire *particulière* ne s'attaque pas à cet aspect que l'auteur argumente pour une grammaire *universelle* :

c'est une propriété essentielle du langage que de nous fournir le moyen d'exprimer un nombre indéfini de pensées et de réagir de façon appropriée dans une série indéfinie de situations nouvelles. La grammaire d'une langue particulière doit dès lors être complétée par une grammaire universelle qui rende compte de *l'aspect créateur* de l'acte linguistique et formule les régularités profondes qui, étant universelles, sont omises dans la grammaire elle-même (...). [Chomsky, 1964, p. 16] (nous soulignons).

C'est avec cette conception du langage, où des phrases peuvent être créées *ad libitum*, que l'auteur argue pour une grammaire *universelle*. Cette conception du langage se retrouve dans la terminologie employée. Ainsi le verbe *engendrer* devient un mot clé, on pourra ainsi lire « une grammaire engendre une phrase » (*ibid.* p. 19), et la grammaire que vise l'auteur est dite *générative*.

3.4.2.c Définition

C'est avec la compréhension forte de grammaire générative, héritée de nos deux derniers points, que nous souhaitons éclairer le couple de concepts *compétence* et *performance*. En introduction nous pouvons préciser que l'ensemble conceptuel que nous venons d'effleuré, entre grammaire, niveaux linguistiques et génération de phrases, est utile pour étudier le langage

74 Par exemple : « L'utilisation normale du langage repose d'une façon essentielle sur cette absence de limite, sur le fait que le langage contient des dispositifs aptes à engendrer des phrases d'une complexité arbitraire. » [Chomsky, 1967, p. 129].

75 De façon synthétique, la théorie de Saussure investit le langage avec un *système de signes* (les deux faces du signe Saussurien), tandis que Chomsky l'investit avec un *système de règles* (ce qui permet d'attaquer l'aspect créatif du langage). Par exemple : « modern linguistic is much under the influence of Saussure's conception of *langue* as an inventory of elements (...) and his preoccupation with systems of elements rather than the systems of rules which were the focus of attention in traditional grammar and in the general linguistics of Humboldt » [Chomsky, 1964, p. 23].

théoriquement, et qu'il l'est moins pour expliquer ce qu'il se passe quand un locuteur utilise le langage pour communiquer oralement. Le couple compétence/performance vient combler cette lacune, il vient expliquer l'acte de communication. Trois paragraphes nous permettront de rendre compte de l'articulation de ces concepts. Le premier concerne les liens avec la langue et les deux derniers concernent les relations qui se tissent.

D'abord, il faut constater qu'une personne qui parle ou qui comprend une langue possède, inévitablement, une *connaissance* de la langue. Et de la même manière, il faut relever que pour arriver à une manifestation physique, sonore par exemple, de la langue, cette connaissance doit être utilisée effectivement. La *compétence* est alors la connaissance d'une langue par un locuteur-auditeur ; la *performance* est l'utilisation effective de cette langue par un locuteur-auditeur.

Ensuite, la compétence n'est pas directement accessible à l'observateur, elle est une *réalité mentale* présente chez le locuteur-orateur. Le fait qu'une personne maîtrise une langue se retrouve dans la cognition de l'Homme, sous une forme mentale. À l'inverse, l'usage de la langue se manifeste dans un espace-temps donné, la *performance* est en lien avec une *situation concrète*, elle est observable, on peut mesurer certaines propriétés (e.g. la vitesse de locution).

Enfin, et cela découle du dernier point « d'observabilité », la *compétence* est *sous-jacente* à un comportement, elle ne peut être saisie directement, tandis que la *performance* fournit des données linguistiques.

La citation ci-dessous permet de résumer ces deux derniers points.

Le problème, pour le linguiste aussi bien que pour l'enfant qui apprend la langue, consiste en ceci : déterminer, à partir des données de la performance, le système sous-jacent de règles qui a été maîtrisé par le locuteur-auditeur et qu'il met en usage dans sa performance effective. De ce fait la théorie linguistique est mentaliste, au sens technique de ce mot, puisqu'elle s'attache à découvrir une réalité mentale sous-jacente au comportement effectif. [Chomsky, 1965, p.13].

La maîtrise d'une langue, sa connaissance, s'envisage comme l'acquisition d'un système de règles, « qui détermine et la forme phonétique de la phrase et son contenu sémantique intrinsèque » [Chomsky, 1967, p. 125] ; et l'action de parler par un locuteur appelle l'usage de ce système de règles. La *performance* fournit des données au linguiste, qui résulte de l'usage de la *compétence* sous-jacente.

3.4.2.d Articulation : grammaire, compétence et performance

Il faut préciser que ces concepts restent des idéaux. C'est-à-dire que la compétence ne détermine pas *entièrement* la performance. Il y a un ensemble de facteurs qui viennent expliquer la performance propre à la situation. La tâche du linguiste est de discerner parmi ces différents facteurs la compétence et de s'attacher à l'éclaircir : « Afin d'étudier une langue, il faut tenter de dissocier un ensemble de facteurs divers, qui interfèrent avec la compétence sous-jacente pour déterminer la performance effective. » [Chomsky, 1967, p.126].

Dans cette voie, la théorie linguistique considère la grammaire comme un modèle de la compétence. L'éclaircissement de la compétence se fait donc par un démantèlement de la grammaire intégrée par un locuteur-auditeur ; par un démantèlement de ce qui lui permet de faire des phrases avec une sémantique voulue.

Nous pouvons alors dessiner le schéma suivant, qui illustre les relations entre Compétence, Performance et grammaire.

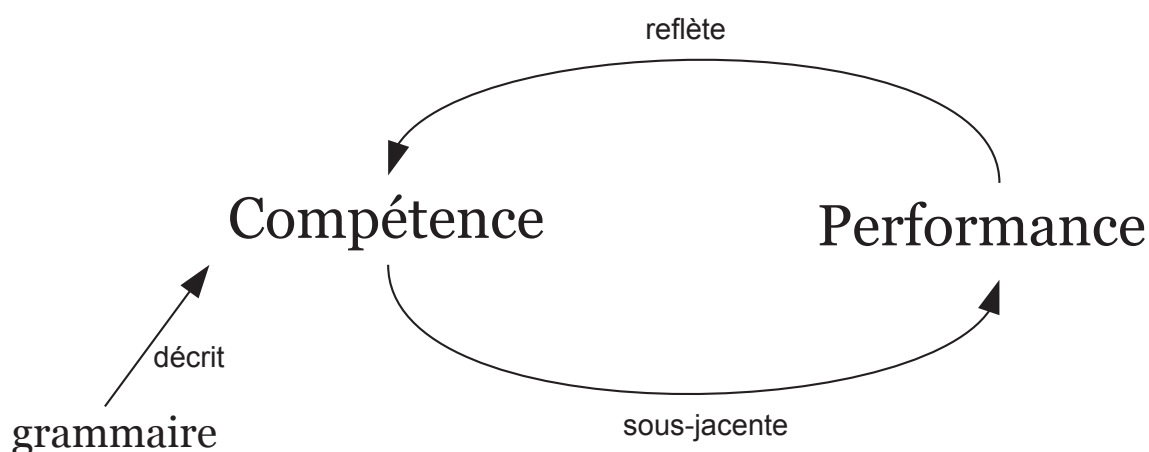


Fig. 11 - Relations entre performance, compétence et grammaire

Enfin, nous souhaitons apporter un dernier point sur l'articulation des deux concepts. Une erreur importante serait de dissocier ces deux concepts, et d'attribuer l'action de parler à la performance, et quelque chose comme l'apprentissage d'une langue à la compétence. Ces concepts fonctionnent ensemble, ils s'éclairent mutuellement. Ils sont tout deux impliqués quand quelqu'un parle, écoute, écrit, lit, dès qu'une personne utilise une langue.

Afin de préciser l'articulation ajoutons deux citations, une première propre au langage, et une seconde qui ouvre aux sciences cognitives.

A modern generative grammar seeks to determine the mechanisms that underlie the fact that the sentence I am now producing has the form and

meaning it does, but has nothing to say about how I chose to form it, or why.
[Chomsky, 1996, p. 26]

The distinction [led by this couple of concepts] is crucial. It is basically the difference between a cognitive system that stores an infinite array of information in a finite mind/brain, and systems that access that information to carry out the various actions of our lives. It is the distinction between knowledge and action—between competence and performance, in standard technical usage. (*Ibid.* p.26).

Cette distinction s'avère fondamentale. Dans [Chomsky, 1996] l'auteur argue que c'est avec celle-ci, en toile de fond, que la révolution cognitive du XXe a pu se produire.

3.4.3 Parentés

La parenté essentielle entre ces concepts et le couple *potentiel/actualisation* vient de l'articulation entre quelque chose qui est abstrait et une autre qui est observable. Ceci est en effet présent dans les deux concepts du langage. Chez Saussure la Langue s'opère à un niveau psychique, entre association de concepts et images acoustiques (signifié et signifiant), tandis que la Parole concerne le côté exécutif du psychique jusqu'à la production sonore par le locuteur. Chez Chomsky cette distinction se retrouve de façon plus marquée. La Compétence est une *réalité mentale*, elle relève de la cognition de l'Homme, et l'on ne peut l'atteindre directement. À la différence la Performance relève de l'observable. Elle fournit les « données linguistiques » ; c'est l'*emploi effectif* d'une langue.

Dans notre contexte ce rapport est aussi présent : le potentiel est bien *abstrait* tandis que l'*actualisation* est elle perceptible, elle concerne l'écoute du son. Et, de la même manière, il n'est pas invraisemblable de dire que l'*actualisation* fournit les données à l'analyste – pensons par exemple à une analyse musicale réalisée à partir d'une analyse de signal (*e.g.* [Malt *et al.*, 2015]).

Mais la parenté fondamentale ne tient pas seulement dans ce partage de propriétés communes. Il y a en effet pour nous quelque chose de plus fort qui concerne les moyens que développe l'analyste pour investir l'objet de son étude. Dans l'étude de « l'acte de parole » comme dans notre cas, l'analyste est confronté à du sonore et il sait que ce qu'il doit étudier dépasse cette simple manifestation physique. L'analyste doit investir quelque chose qui est sous-jacent (N. Chomsky) à cette manifestation. La parenté fondamentale tient alors dans la définition de deux concepts qui permettent à l'analyste d'avancer dans son étude. Saussure distingue la Parole qui est *subordonnée* à la Langue ; Chomsky la Compétence qui est *sous-jacente* à la Performance. L'analyste (ou le linguiste) est bien confronté à un seul fait qu'il investit, explique, à l'aide d'un couple de concepts qui *fonctionnent ensemble*. La parenté

fondamentale tient, en plus de ces propriétés communes, dans le fait que ces concepts vont de pair.

3.4.4 Éclairer la dynamique du cadre

C'est cette circularité, présente entre les deux concepts de *potentiel* et d'*actualisation*, qui va nous permettre d'éclairer un aspect dynamique du cadre, celui qui à lieu à l'échelle des Modules, où techniques de synthèse et de traitement s'enchevêtrent.

Nous avons vu précédemment que le couple potentiel/actualisation était utile pour investir un Module : l'analyste confronté à un Module doit éclairer le potentiel sonore que le compositeur lui a attaché. Dans cette situation, la musique d'informatique est peut-être particulière dans la mesure où l'analyste est en fait confronté à nettement plus que le *potentiel* du compositeur. En lisant l'implémentation, et surtout en effectuant différents paramétrages d'un Module qui engendre différentes sonorités, l'analyste a accès, et est confronté, à bien plus que le *potentiel* sonore du Module. La différence entre l'ensemble de possibles permis par l'implémentation et le potentiel tient dans *l'utilisation* qui en est faite par le compositeur.

Après avoir vu que le potentiel n'est pas l'ensemble de possibles qu'une méthode de synthèse permet, il nous faut, de surcroît, souligner que le potentiel n'est pas non plus le réservoir sonore de l'implémentation d'un ensemble de techniques, mais bien l'utilisation de cette implémentation.

Pour cibler le potentiel d'un Module l'analyste doit, en parallèle d'une lecture de son implémentation, écouter l'usage qui en a été fait, lire les *actualisations* réalisées. C'est dans cette réunion de l'écoute et de l'implémentation que le couple de concepts potentiel/actualisation vient éclairer un aspect dynamique de notre cadre. L'identification et l'imbrication des Modules doit se faire selon l'utilisation faite par le compositeur. Ce travail est nécessairement dynamique, puisque le Module ne peut se réduire à « du code », il est *habité* ; pour l'investir il faut se confronter aux sons que le compositeur lui a fait engendrer.

La représentation ci-après rend compte de la circularité fondamentale avec laquelle les concepts de potentiel et actualisation prennent sens.

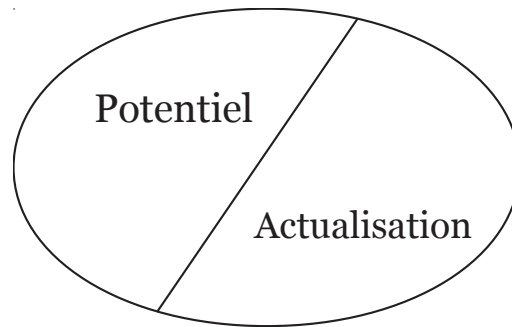


Fig. 12 - Le couple de concepts potentiel et actualisation

La circularité est mise en avant par l'ellipse et par sa division non verticale. *potentiel* et *actualisation* ne se départagent pas selon la verticale, mais partagent un espace commun : ils se définissent l'un l'autre...⁷⁶

⁷⁶ C'est particulièrement F. de Saussure qui nous a mis sur la voie de cette réciprocité. Chez l'auteur en effet les relations entre Parole et Langue sont telles que la Parole est à la fois « le produit et l'instrument » de la Langue.

4 Structure de contrôle

Notre proposition théorique pour investir les artefacts se termine avec le concept de *structure de contrôle*. Ce dernier s'avère utile pour deux raisons. D'abord il permet de finir notre cadre théorique, et ensuite il permet une dernière fois d'éclairer l'aspect dynamique. Nous expliquons par la suite chacune de ces raisons.

4.1 Regroupement de contrôles

Le premier sens d'une *structure de contrôle* nous vient d'Agostino Di Scipio ; on peut le comprendre comme un regroupement de contrôles.

the controls that the composer finds of real significance for creative purposes, are embedded into a control structure which allow him/her to evaluate the synthesis parameters according to higher-level strategies [Di Scipio, 1995a, p. 38].

La structure de contrôle se place ainsi au-dessus des *contrôles* de notre cadre. Elle est au plus proche de ce que Di Scipio appelle des *stratégies de haut niveau*, ce que nous avons aussi appelé les idées musicales. Pour envisager cette structure on peut simplement comprendre que les contrôles qui résultent de la composition sont très nombreux – corrélativement à la densité de l'artefact. La *structure de contrôle* englobe alors ces différents contrôles avec lesquels la composition s'est effectuée. Ce regroupement s'avère essentiel puisque l'action du compositeur ne se fait pas de façon isolée sur un contrôle, tel un interrupteur, mais bien sur un ensemble de contrôles. Ces contrôles peuvent en effet être indépendants les uns des autres, et c'est en ayant intégré toutes ces interdépendances que l'action compositionnelle s'exerce.

Pour illustrer ces interdépendances on peut par exemple penser à un Module de transposition, qui intègre un contrôle sur la taille de la fenêtre audio à enregistrer (buffer). Le taux de transposition à effectuer est alors à considérer avec ce contrôle de fenêtre audio⁷⁷. Aussi, à plus grande échelle, ce besoin de regroupement se comprend dans le cas de chaînes de traitement récurrentes, où la même suite de Module de traitement avec le même paramétrage est appliqué⁷⁸.

77 Plus la transposition est grave plus la fenêtre doit être grande, et inversement.

78 Dans *Jupiter* la chaîne de traitement Reverb infinie vers transpositeur est récurrente par exemple.

La structure de contrôle rend compte ainsi de l'ensemble des contrôles avec lesquels le compositeur a réalisé sa musique et des interdépendances entre contrôles. Dans notre cadre cette structure est donc au plus haut niveau d'abstraction, elle se place donc au sein de notre représentation au plus proche de l'Homme.

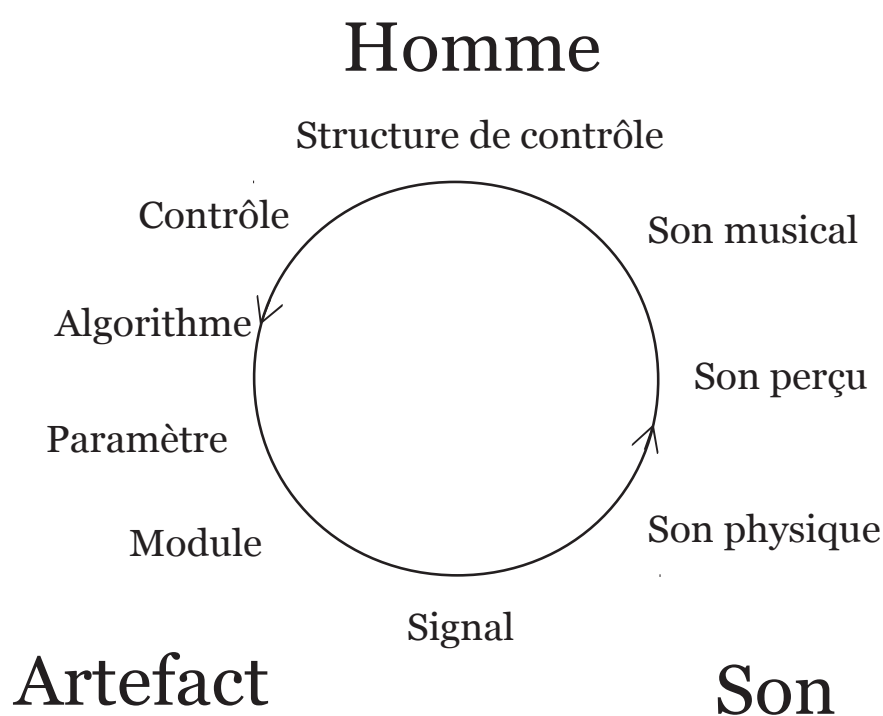


Fig. 13 - Positionnement de la structure de contrôle au sein du cadre conceptuel

4.1.1 Utilisations de contrôles

Le second sens que nous donnons à la structure de contrôle vient du couple de concept Compétence/Performance. L'éclaircissement que nous apportons vise le même niveau d'abstraction que le premier, il ne s'agit pas de cibler des structures d'un niveau plus faible. En liminaire il nous faut alors préciser que nous utilisons ici le couple Compétence/Performance en dehors de la parenté que nous avons faite avec le couple *potentiel/actualisation* – qui concerne lui un niveau plus bas, celui des *paramètres*. On peut en effet comprendre que même s'il réside des liens de parenté, le couple Compétence/Performance déborde amplement du second.

Le second sens de la structure de contrôle concerne aussi l'ensemble des contrôles avec lesquels la musique s'est réalisée. Ce serait une mauvaise direction de voir ces contrôles comme des points d'accès qui restent intacts après la composition, qu'il resterait à étudier. Ces contrôles peuvent bien s'éclairer avec des lectures de l'artefact – de façon ascendante – mais il nous paraît essentiel de ne pas appliquer la compréhension qui résulterait de ces lectures sur les *contrôles* de

l'artefact. C'est ce que nous souhaitons éviter avec ce second sens de la *structure de contrôle*. Une telle structure dépasse un regroupement de contrôles – à la fois peu signifiant musicalement et trop technologiquement – pour cibler *l'utilisation* qui en a été faite par le compositeur. La *structure de contrôle* résulte de l'utilisation des groupements de contrôles qui a été fait durant la composition.

Cette importance de l'utilisation – que nous avons aussi vue au niveau des Modules – nous vient de la différence entre Compétence et Performance. Dans [Chomsky, 1996, p. 26] il est dit que l'étude des mécanismes qui permettent de donner du son et du sens à une phrase, n'a, en revanche, rien à dire quant à la formulation de cette phrase. Il faut alors différencier un mécanisme, la Compétence, de *l'utilisation* de ce mécanisme, la Performance⁷⁹.

La structure de contrôle n'est donc pas ce qu'un ensemble de contrôles permet (*affordance*), mais l'ensemble des utilisations qui ont été faites par le compositeur. Soit, au fond, ce que ces contrôles représentaient dans l'acte compositionnel.

C'est à l'issue du concept riche de structure de contrôle que nous apportons le premier élément de réponse à notre problématique. L'analyse d'une pièce de musique d'informatique peut s'aider de l'artefact en mettant au jour sa *structure de contrôle*, *i.e.* l'ensemble des *contrôles* avec lesquels la musique s'est réalisée, s'est pensée. La *structure de contrôle* vient préciser que les groupes de *contrôles* visés ne sont pas ceux du *patch* ou du *code*, mais ceux qui ont permis le développement d'idées compositionnelles.

79 C'est la distinction chère à l'Intelligence Artificielle : « to know what knowledge is used in a task does not suffice for explicating knowledge. Stipulations as to the 'what' of knowledge (competence model) are inseparable from the 'how', *i.e.* from the procedures by which knowledge is activated and used. » [Laske, 1990], où des modèles de Compétence et de Performance sont appliqués à la composition musicale.

Chapitre V

Analyse Musicale, positionnement

La musique est faite pour être entendue

Pierre Schaeffer

La musique est faite pour être lue

Celestin Deliège

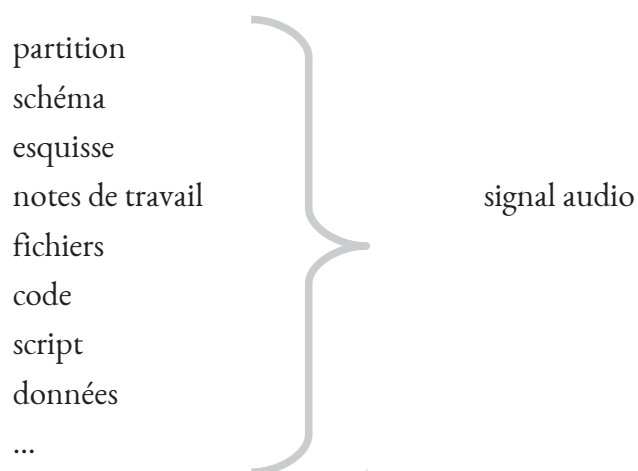
Résumé

Ce chapitre est un positionnement théorique dans le champ de l'analyse musicale. Il rend compte de la méthode d'analyse que nous utilisons dans les chapitres suivants. C'est aussi dans celui-ci que nous apportons la réponse finale à notre problématique.

Dans un premier temps, en pointant les objets physiques utilisés en analyse, nous proposons une répartition des méthodes d'analyse des musiques électroacoustiques. Ensuite nous détaillons celles qui nous sont proches, c'est-à-dire celles qui partent d'une prise en compte forte des traces technologiques. Dans un deuxième temps nous mettons en lumière une spécificité de la musique d'informatique, qui ne nous semble pas avoir été substantiellement prise en compte dans les théories analytiques existantes. Enfin, nous apportons le second élément de réponse à notre problématique. Nous donnons la voie que nous suivrons pour analyser la pièce de Philippe Manoury. Notre proposition se base sur une pratique déjà connue, intitulée, par son concepteur Michael Clarke, *Interactive Aural Analysis*. La nouveauté que nous apportons se trouve sur le plan théorique. Alors que la précédente se conçoit du côté de la perception, nous proposons de partir de la perception et des connaissances issues de la compréhension de l'artefact pour analyser la musique.

1 Analyser la musique électroacoustique

L'analyse de la musique électroacoustique a un passé riche de plus d'un demi-siècle. Durant cette période, nombre de théories, pratiques et outils analytiques – parfois en correspondance avec les avancées technologiques – ont vu le jour. De prime abord, nous proposons d'éclairer cette pratique à l'aide du schéma ci-dessous. Ce dernier doit être compris comme une première base de travail pour approcher l'analyse de ces musiques.



Ce schéma nous permet de différencier deux directions dans l'analyse. La première consiste à partir du signal audio de la pièce, dans le sens où nous l'avons défini⁸⁰. Il s'agit alors de partir de la *perception* physique de la pièce pour l'analyser. À la différence, la seconde proposition part elle de « ce qu'il y a en amont du signal », soit, le plus largement possible, de toutes sortes de documents qui ont contribué, de près ou de loin, à la composition de la pièce. Il peut ainsi être question de fragments de partitions, d'esquisses, de schémas, de codes informatiques, de scripts, de fichiers audio, de patches, de notes de travail ...

Cette vue approximative laisse bien évidemment penser à la tripartition sémiologique, amenée par Jean Molino [Molino, 1975] et grandement développée et disséminée par Jean-Jacques Nattiez ([Nattiez, 1975] réédité en 1987). En effet, la pratique de l'analyse qui utilise des « objets » émanant de la composition, correspond à une analyse du poïétique, et une analyse basée sur l'écoute du signal correspond à une analyse dite esthétique. Nous n'avons cependant pas intégré ces termes connus dans notre schéma pour la raison qu'ils sont très chargés théoriquement. Or, justement, des pratiques analytiques peuvent partir de ces objets sans nécessairement s'inscrire théoriquement dans cette sémiologie. En effet, comme nous le verrons,

⁸⁰ C'est-à-dire en tant que représentation de ce que l'on perçoit (cf. Chapitre II 2.2.1)

il est tout à fait possible d'utiliser des « traces du poïétique » sans pour autant souscrire à l'approche poïétique de la sémiologie⁸¹.

En somme, un avantage de partir des objets physiques utilisés pour éclairer différents types d'analyse, se trouve dans le fait que l'ordre obtenu est particulièrement intelligible. Par exemple l'analyse de Pierre Couprie sur les *Trois rêves d'Oiseau* (1971) de François Bayle ([Couprie, 2003]), est faite à partir du signal, tandis que l'analyse de *Stria* faite par [Meneghini, 2007] est elle faite en amont du signal, à partir du code Music N essentiellement.

Notons que cette différenciation (« à partir » vs. « en amont » du signal) est présente dans un récent travail de Pierre Couprie [Couprie, 2016]. Après avoir représenté « les principales théories qui ont été élaborées par les compositeurs pour la composition, l'interprétation ou l'analyse musicale », l'auteur témoigne de deux tendances dans les méthodes d'analyse. Les premières sont « fondées sur la perception auditive du support audio », et les secondes utilisent « les sources de la création rendues disponibles par le compositeur » (p. 160).

Enfin, précisons que cette différenciation prend sens à un niveau théorique, dans le positionnement de l'analyste. Car, comme le souligne la citation suivante extraite d'un travail d'analyse réalisé « en amont du signal » – sur lequel nous reviendrons –, l'analyste est inévitablement en contact avec l'écoute de musique :

Notre premier objet, chronologiquement, c'était *De Natura*⁸², c'était la musique. Il avait fallu d'abord "l'interroger" par l'écoute, de façon à ensuite, par des questions si possible pertinentes, aiguiller, stimuler le compositeur.
[Thomas *et al.*, 1982]

Dans le présent grand titre nous donnons des exemples de méthodes s'inscrivant soit en amont soit à partir du signal. Évidemment nous détaillons surtout les premières, dans lesquelles notre travail s'inscrit.

1.1 Analyser à partir du signal

L'analyse à partir du signal de la pièce, qu'il soit analysé par voie technologique ou simplement lu, nous semble être la tendance la plus courante. Les théories développées dans

81 Plus précisément notre éloignement avec cette tripartition sémiologique se trouve dans la zone de flou qu'elle amène en distinguant un processus poïétique d'un processus esthétique. Où en effet, dans la musique instrumentale, commence le processus esthétique ? Commence-t-il dès lors qu'elle est lue par l'interprète ou dès lors qu'il y a perception chez un auditeur ? J.-J. Nattiez, pleinement conscient de cette « zone de flou », considère ces deux cas (cf. [Nattiez, 1987, p. 99]). Ce problème connu qui concerne la délimitation d'un niveau neutre, d'une « matérialité de la trace symbolique » (*ibid.* p34), se retrouve avec une plus grande résonance pour la musique électroacoustique (cf. [Dahan, 2005, p. 55] par exemple).

82 *De natura Sonorum* (1975), Bernard Parmegiani

cette perspective sont nombreuses et toujours en évolution (voir par exemple [Emmerson *et al.*, 2016] ou encore les publications du congrès EUROMAC de 2014 dans [Marty *et al.*, 2015]). Il est donc seulement question ici d'approcher un certain contexte propice à ce développement.

1.1.1 Une musique à écouter

Expliquer pourquoi l'analyse des musiques électroacoustiques se fait essentiellement par son signal peut paraître, de prime abord, quelque peu étonnant. En somme cela pourrait revenir à expliquer pourquoi l'analyse de la musique occidentale se fait avec sa partition. Il nous semble cependant que ces évidences cessent dès lors que l'on change d'angle de vue, ce que nous essaierons de montrer.

L'évidence de cette approche pour l'analyse se trouve dans ce que l'on pourrait appeler l'embryon des musiques électroacoustiques, la musique concrète, amenée par Pierre Schaeffer. Au centre de celle-ci se trouve le concept phénoménologique, cher à P. Schaeffer, de l'*écoute réduite* : le son doit être perçu sans considérer sa cause, mais uniquement dans l'acte de perception.

De l'écoute réduite, à la fois concept et pratique, découle tout un nouveau champ pour l'analyse musicale, qui consiste à réfléchir à des moyens pour analyser la musique à partir de son écoute, sans chercher inexorablement à saisir quelque chose du processus compositionnel (du poétique). Cette position dans la pratique de l'analyse est bien nouvelle, comme en témoigne François Delalande dans l'introduction de son recueil [Delalande, 2013]. Après avoir qualifié de « période heureuse » un moment où des grandes théories analytiques sont apparues (le modèle Schenkerien, la *Generative Theory of Tonal Music*, la *Set theory*, l'analyse paradigmatique de Ruwet et Nattiez) l'auteur voit, avec les musiques électroacoustiques, « un nouveau paradigme [qui] sème le désordre, si ce n'est la révolution⁸³ » (*Ibid.* p. 7).

Sans partition, et avec le concept d'écoute réduite, les moyens théoriques issus de l'analyse des musiques électroacoustiques, partent tous d'un principe fort, celui de considérer le *signal audio* de la pièce :

L'objet qu'il s'agit d'analyser, c'est-à-dire une pièce ou un extrait donné sous sa forme sonore, a bien une certaine matérialité : c'est un signal, fixé sur un support, qui devient une onde acoustique [Delalande, 2013, p. 10]

83 En référence à Tomas Khun

1.1.2 Entre signal physique et écoute subjective

Depuis ce positionnement (~1970), que l'on pourrait symboliser par la phrase de P. Schaeffer placée en exergue, c'est tout un arsenal de théories et de méthodes qui s'est développé. Évidemment face à cette ampleur, le fait de dire que ces méthodes et théories partent du signal, s'avère quasi insignifiant. Un autre pas à faire serait par exemple d'éclairer le rapport entretenu avec le signal audio. On pourrait alors différencier des pratiques où le signal est utilisé comme une représentation de l'écoute – où des analyses technologiques du signal sont alors effectuées (sonogrammes et descripteurs audios par exemple) –, de pratiques où le signal est un support de l'écoute, nécessaire pour naviguer dans l'écoute. On retrouve par exemple cette organisation dans [Marty, 2015], avec une dimension verticale : en bas sont présents les sonogrammes et descripteurs, puis, au-dessus, et successivement, les célèbres théories d'analyses : la typomorphologie de P. Schaeffer, la spectromorphologie de Denis Smalley, et enfin la musique envisagée comme « expérience humaine », avec les conduites d'écoute de F. Delalande.

1.2 Analyser en amont du signal

Même si les musiques électroacoustiques se font sans partition, il est tout de même possible de les analyser avec des documents situés en amont du signal. C'est par exemple ce qui a été fait en 1982 par deux chercheurs du Groupe de Recherche Musicale (GRM), Philippe Mion et Jean-Christophe Thomas, dans un ouvrage qui fait date, [Thomas *et al.*, 1982]⁸⁴. L'ouvrage est consacré à la célèbre pièce *De Natura Sonorum* (1975) de Bernard Parmegiani, pour bande, qui est analysée à partir de différents documents provenant de la composition (esquisses et partitions d'écoutes), et aussi à partir d'interviews réalisés avec le compositeur même. En suivant la célèbre tripartition sémiologique, ce travail s'inscrit donc dans une étude du poïétique.

Nous donnons ci-après un aperçu de deux méthodes d'analyse qui se font à partir de documents situés en amont du signal, et qui concernent davantage la technologie que la précédente sémiologie – qui est résolument générale. Il s'agit de la génétique musicale, puis de la *faktura*.

1.2.1 Génétique musicale et informatique

La génétique musicale est un domaine à part entière qui déborde amplement de nos travaux. Elle se dessine par exemple autour de domaines connexes comme la philologie et l'herméneutique. Afin de contextualiser l'application de cette dernière sur la musique d'informatique, et au risque de paraître impertinent, nous souhaitons dans un premier temps donner synthétiquement ce qui nous semble être la grande ligne de la génétique musicale.

84 En plus de l'analyse réalisée par ces deux chercheurs, l'ouvrage intègre un chapitre de Jean-Jacques Nattiez qui présente des réflexions théoriques sur l'analyse poïétique.

1.2.1.a Génétique et écriture

Nos propos sont issus d'une lecture d'un récent ouvrage collectif qui réunit plusieurs applications de la génétique à la musique [Donin, 2015]⁸⁵. À la lecture de l'introduction qui contextualise la pratique analytique, il ressort – synthétiquement donc – deux points essentiels. D'une part la génétique prend forme à partir d'une très forte considération sur un *concept d'écriture* – et de celui-ci découle son « amour pour le texte » (*ibid.* p. 12) –, et d'autre part la génétique s'intéresse, tautologiquement, à la gestation de l'œuvre.

Ces deux points étant inévitablement corrélés : c'est par le principe même de l'écriture – disons, « qui fige le temps dans lequel elle se déroule » – que la génétique peut « remonter » la gestation de l'œuvre :

en suspendant le temps, en offrant des outils d'organisation et de hiérarchisation des contenus, en permettant le retour en arrière, la relecture, le repentir, la réécriture, *l'écrit* constitue une *technologie cognitive* particulièrement adaptée à des processus de création complexe (*ibid.* p. 11, nous soulignons)

Du point de vue de la pratique, la génétique musicale se réalise donc avec des esquisses émanant de la composition. Historiquement ces dernières sont essentiellement des manuscrits utilisant la notation musicale, ou encore des partitions manuscrites. Mais, à l'aune du XXI^e, comme nous le verrons ci-après, avec l'usage de technologies, ces « esquisses » peuvent être amenées à changer.

1.2.1.b Musique d'informatique et génétique musicale

La génétique musicale n'est en effet pas restée hermétique à la musique d'informatique. En effectuant un pas conceptuel, que nous jugeons assez important, il est possible de l'appliquer à la musique d'informatique. Néanmoins, en raison certainement de ce pas conceptuel, les applications restent rares. De notre connaissance les premiers travaux ont été faits par Laura Zattra⁸⁶, [Zattra, 2003].

Le pas conceptuel nous semble important puisque dans le même mouvement qui consiste à envisager l'acte d'écriture avec l'informatique, il faut, de même, considérer les traces qui résultent de la poïétique comme du « texte ».

85 Dont certains chapitres sont issus d'un colloque intitulé "Genèse musicale, Méthodes et Enjeux", tenu en 2008

86 Par exemple, dans le panorama des méthodes pour l'analyse des musiques électroacoustiques établi par l'auteur [Zattra, 2005], dans le paragraphe « génétiques musicales », à part ses propres travaux, seul celui de [Lorrain, 1980] est cité.

Les œuvres [de computer music] sont en général réalisées dans de grands centres de recherche et production, et montrent, à l'analyse, une prédilection pour l'idée "d'artisanat" qui se réalise à travers *l'écriture*, une action caractérisée par la présence de *textes*, au sens large du terme. [Zattra, 2015, p. 214].

L'application de la génétique à la musique d'informatique nécessite donc inévitablement un « élargissement du concept de texte » (p. 220). L'ordinateur est alors vu comme « une source pour l'étude du processus/traduction entre l'univers musical du compositeur et l'univers numérique, il produit des « textes » qui schématisent et opérationnalisent des choix compositionnels » (*Ibid.* p. 220).

À l'aide de ces textes, l'analyste va donc pouvoir mettre au clair les choix compositionnels, et par la même relever le « processus de constitution d'une œuvre » (p. 220).

1.2.2 La *faktura*

En 2003 Marc Battier apporte un air nouveau dans l'étude des pièces de musique électronique, avec des idées venant d'un courant artistique Russe du début du XXe, le Constructivisme. Son approche est présentée dans un article qui fait date, *A Constructivist approach to the analysis of electronic music and audio art—between instruments and faktura*, [Battier, 2003], si bien que beaucoup des travaux s'appuyant sur les « technologies de production sonore » pour éclairer une pièce s'y réfèrent. Citons à titre d'exemple [Baudoin, 2009], [Dufeu, 2010], [Bergsland, 2011], et [Bonardi, 2013a]. Ci-après nous décrivons l'article de M. Battier afin de saisir l'outil analytique proposé.

Cet outil analytique clé, la *faktura*, n'est pas issu de réflexions portant exclusivement sur l'usage de l'informatique, mais sur l'utilisation de technologies de production sonore tous azimuts⁸⁷ : toutes les technologies du XXe sont ainsi concernées, électrique, électromécanique, électroacoustique, électronique et enfin audionumérique [Battier, 2003, p. 251].

1.2.2.a Présentation

En liminaire du concept de *faktura* se trouve une idée essentielle : technologie, technique et pensée esthétique ne sont pas sans liens :

since the beginning of the XX, technology, artistic technique and aesthetic thoughts have been intertwined (*ibid.* p. 251)

87 Une volonté de dépasser certaines frontières est ainsi présente ; l'auteur parle d'art audio.

À partir de cette idée, l'auteur utilise le Constructivisme pour mettre l'emphase sur les interactions qui se tissent, lors d'un processus créatif, entre technologie, technique et style⁸⁸. La proposition théorique amenée consiste alors à étudier le processus de production technologique d'une pièce de sorte à éclairer le style de l'artiste.

1.2.2.b *Factura* et Constructivisme

La *faktura*⁸⁹ prend sens avec un autre concept hérité du Constructivisme Russe, la *tectonic*. Ce dernier « désigne la portion de style qui intervient du dehors et qui s'impose de soi-même à l'artiste » (*ibid.* p. 251). Il permet de distinguer l'action de l'artiste, de ce qui s'impose à lui, ce qui « préside à l'élaboration des pièces »⁹⁰ soit, un vaste ensemble où se mélangent matériaux et styles.

the theoretical effort of the Constructivist was aimed at defining the relation between the act of creation and its materials. [Battier, 2003, p. 251]

C'est donc à partir de cette distinction que le second concept du Constructivisme, la *faktura*, est défini. Il s'agit de cibler ce moment où l'artiste – "either a sculptor, painter, poet, musician or audio artist" (*ibid.* p. 251) – va agir sur ce vaste mélange de matériau et de style. Le moment où il va « sélectionner consciencieusement » des matériaux, et les « utiliser rationnellement » (p. 251) ... le moment où il y a transformation :

Faktura is the category with which an artist transforms material (*ibid.* p. 251)

Et cette transformation, inévitablement, se fait par l'intermédiaire de la technologie :

[Faktura] focuses on the nature of (...) the how the artist transforms and adapt the technology (*ibid.* p. 251).

Avec ce concept il s'agit donc de cibler un moment dans le processus de création, où l'artiste sélectionne puis transforme des matériaux.

88 Une telle approche est donc en lien avec les travaux menés sur la *techné* et la musique électroacoustique, cf. [Di Scipio, 1995a], [Di Scipio, 1995b], [Manning, 2006].

89 Il faut relever que la définition de ce concept n'est pas évidente. D'abord, comme le souligne l'auteur "a leap from art theory to music is far from obvious", et ensuite, à plus forte raison, ce concept Constructiviste prend sens avec deux autres, nommés *tectonic* et *construction*. Or, le présent article, faute de place, rend compte essentiellement du premier et très peu du second. Ainsi, dans la présentation que nous faisons de la *Faktura*, il n'est pas impossible qu'il réside une certaine « appropriation musicologique ».

90 [Baudoin, 2009, p. 77]

1.2.2.c Analyse musicale et *faktura*

C'est après une explication de la *faktura* que le domaine musical est questionné. L'auteur se pose alors la question suivante, non sans liens avec notre problématique : « comment analyser les artefacts sonores ? »⁹¹. La réponse se fait en deux temps. Dans un premier il s'agit d'acquérir le savoir présent dans les systèmes de production sonore ; dans un second d'étudier la façon dont ces systèmes sont intégrés à (i) la composition (« the symbolic manipulation ») et à (ii) la manifestation sonore (« the physical manipulation »)⁹² (*ibid.* p. 252).

Enfin, la finalité de la *faktura* est abordée à la fin de l'article. À l'issue du processus sous tendu par la *faktura*, en sélectionnant et en s'appropriant la technologie de leur temps, les artistes créent en fait une *technique* qui leur est propre. Dans cette technique, fruit du travail artistique, singulier à une époque et à un artiste, il est possible de lire des « traces stylistiques ». La *faktura* a donc pour fin de mettre au jour ce style imbriqué dans la technique. De là, il doit être possible d'éclairer « la contribution personnelle du compositeur », de le situer vis-à-vis d'écoles esthétiques auxquelles ils se rapportent ou non (*ibid.* p. 254).

91 (Notre traduction) « in what way may sonic materials, an artefact of artificial sound art, be analyzed ? » (*ibid.* p. 252)

92 On retrouve ici deux points essentiels vues précédemment : les artefacts (i) contiennent des connaissances, et (ii) se structurent en différents niveaux, ici du physique au symbolique (cf. Chapitre I 1.3 et Chapitre II 4)

2 Spécificité de la musique d'informatique

Nous avons vu précédemment que l'analyse de la musique d'informatique pouvait se faire « en amont du signal ». Deux approches ont été présentées : l'approche génétique [Zattra, 2015] et la *faktura* [Battier, 2003]. Au regard de ces approches analytiques, il nous semble que la musique d'informatique possède une particularité conséquente pour l'analyse, qui, de notre connaissance, ne semble pas avoir été nettement prise en compte. Nous ciblons ici cette spécificité en deux temps. D'abord nous montrons au travers d'écrits pionniers qu'il réside une certaine difficulté de prise en compte de la musique d'informatique, et ensuite nous expliquons cette spécificité.

2.1 Difficulté de prise en compte

La question de la musique d'informatique et de son analyse a été amenée selon nous par Marco Stroppa en 1984 [Stroppa, 1984]. Les questions apportées dans ce texte sont ensuite traitées par Jean-Claude Risset qui apporte des éléments de réponses en 2001, [Risset, 2001]. L'article de M. Stroppa s'avère en fait symptomatique d'une certaine difficulté de prise en compte, dans une démarche analytique, de cette musique d'informatique. Il est symptomatique puisqu'il est en fait issu d'une impossibilité : celle pour l'auteur d'analyser la célèbre pièce, *Sud* (1985) du même Jean-Claude Risset.

Nous donnons ci-après trois points qui témoignent globalement, en cette époque, d'une difficulté de prise en compte de la musique d'informatique.

2.1.1 Du code

En premier lieu, et même si cela s'avère tautologique, il faut relever que la musique d'informatique engendre des « traces codées » [Risset, 2001, p. 16]

Par une cascade de conséquences, ce constat nous amène à deux considérations, qui ont déjà été approchées dans la première partie. Ces traces :

- peuvent être lues à la condition *sine qua non* de connaître le langage de programmation utilisé (cf. Chapitre I 1.3)
- sont le fruit d'une activité de *design* ; elles intègrent de fortes idiosyncrasies (cf. Chapitre I 1.4).

Ceci vient souligner la difficulté de lecture de ces traces codées. D'une part, il est nécessaire de connaître, si ce n'est de maîtriser, un langage de programmation – et rappelons que ces langages évoluent inévitablement avec la technologie –, et d'autre part il s'agit de lire des traces fortement

idiosyncratiques. Cette dernière difficulté est moins évidente que la première. Elle nous amènerait à faire un *distinguo* entre une lecture linguistique, issue d'une langue, et une « lecture computationnelle », issue d'un travail de design dans un langage de programmation.

Cette difficulté de lecture est par exemple présente chez J.-C. Risset, qui parle de trace cryptique :

Même pour qui est familier avec le langage, les partitions MUSIC V sont cryptiques, les processus y sont décrits de façon quasi-atomique. Jean-Claude Risset dans [Lorrain, 1980]

cryptic traces ... need proper explanation (J.-C. Risset dans l'introduction de [Licata, 2002]).

De même on retrouve ceci chez A. Di Scipio, dans le cadre d'un travail d'analyse sur la pièce *Contours* (1982) de J.-C. Risset :

I scrutinized many music v code listings that the composer had written for this piece, and that he had kindly made available to my investigation [Di Scipio, 2000, p. 2].

Enfin, cette difficulté de lecture se retrouve dans [Stroppa, 1984] de façon criante. Il serait tentant de penser que l'abandon du projet d'analyse de *Sud* aurait pu être évité si l'auteur avait eu à sa disposition le code de Music N – ce que Jean-Claude Risset suggère dans sa réponse [Risset, 2001]. Mais cela est vain. À l'époque de l'article (1983), et cela est fondamental, où l'ordinateur n'est pas démocratisé, ces traces sont envisagées comme des « données opératoires » et elles ne peuvent nullement être utiles pour l'analyse musicale :

[ces données opératoires] sont liées à une machine et à des programmes spécifiques. Totalement incompréhensibles à la majorité, elles s'adressent seulement aux super-spécialistes. [Stroppa, 1984, paragraphe 3]

2.1.2 Musique sur bande

Le second point qui témoigne d'une difficulté de prise en compte concerne l'appellation même de cette musique. Dans ces années pionnières, la musique d'informatique est aussi désignée par *musique sur bande*. Le fait de la nommer ainsi freine, inévitablement, la prise en compte de ce avec quoi elle a été conçue, soit, disons ici, son code informatique.

C'est aussi présent chez [Stroppa, 1984]. Après une introduction avec des considérations sur la musique électronique, l'objet de l'article se « borne à ne considérer que cette *musique pour bande* qui essaie de rester en contact avec l'évolution de la pensée musicale contemporaine » (*Ibid.* paragraphe 2, nous soulignons). Puis, au paragraphe suivant, où il est question des possibilités de l'analyse musicale soit à partir des données opératoires soit à partir des esquisses perceptives, c'est bien, inévitablement, d'ordinateur dont il est question :

en essayant de garder [les principes de l'analyse musicale traditionnelle] et de les appliquer à la musique pour ordinateur, on découvre une réalité et des problèmes tout à fait nouveaux (*Ibid.* paragraphe 3).

Dans la réponse faite par [Risset, 2001], un des points centraux se trouve, selon nous, dans un recentrement des questions posées par M. Stroppa. Pour J.-C. Risset, il n'est plus question de musique électronique, ou encore de musique sur bande, mais bien, comme le précise sans équivoque le titre de l'article, « d'œuvres musicales dont la réalisation fait appel à l'informatique ».

De façon peut être plus étonnante nous retrouvons aussi cette association chez [Lorrain, 1980]. Simplement, le document est intitulé : « Analyse de la bande magnétique de l'œuvre de Jean-Claude Risset "Inharmonique" ». C'est étonnant puisque si ce travail fait date, comme le souligne [Zattra, 2015], c'est justement parce qu'il est un des premiers⁹³ à partir des partitions de Music N, du code, pour éclairer la pièce ... et non du signal audio figé sur la bande.

Aussi, dans une même lignée, on peut remarquer que ce travail n'est pas édité. C'est en effet un rapport Ircam (sans pagination dans la version en ligne), réalisé à « l'intention des stages Ircam pour compositeur » comme le précise Jean-Claude Risset dans la préface.

2.1.3 Ontologie

Cette difficulté de prise en compte du code informatique nous semble pouvoir être expliquée, et de façon nettement plus profonde, en considérant l'ontologie des œuvres musicales. Simplement, si le code restait, à cette époque pionnière, peu pris en compte, c'est que l'on ne le considérait pas comme ayant une quelconque importance vis-à-vis de l'existence de l'œuvre.

93 D'autres travaux ont été faits dans la même période avec une prise en compte du code (comme [Haynes, 1981] ou encore [Gerszo, 1984]), mais l'originalité ici c'est que l'auteur n'est ni le compositeur ni un assistant musical : il est extérieur à la pièce, à l'instar d'un analyste.

C'est par exemple limpide chez M. Stroppa, au début et à la fin du paragraphe 3 où il est justement question de l'analyse de ces musiques :

Une des caractéristiques principales [de la musique sur ordinateur] réside dans son *existence* et sa diffusion exclusives sur bande magnétique. [Stroppa, 1984] (ouverture paragraphe 3) (nous soulignons)

Ni les données opératoires, avec leurs froides dispositions techniques, ni les esquisses graphiques du compositeur ne peuvent ainsi être considérées comme des « partitions ». On a ainsi éliminé tout ce qui existe actuellement. [Stroppa, 1984] (fermeture paragraphe 3)

À notre époque, soit trente ans après l'époque pionnière, et *a posteriori* des importants travaux de reconstruction de pièces alliant musique et ordinateur, tant « sur bande » (e.g. *Stria* ou *Inharmonique*) que « pour électronique en temps réel » (e.g. *Diadèmes* ou *En echo*), il nous semble évident que cette musique d'informatique puisse aussi exister *en dehors* de l'inscription physique de son signal, *en dehors* de la bande.

2.2 Une écriture

La musique d'informatique posséderait donc bien quelque chose de spécifique, qui fait qu'elle serait mal saisie quand elle est envisagée comme une « musique sur bande ». Cette spécificité nous semble avoir été éclairée chez deux auteurs/compositeurs J.-C. Risset et Horacio Vaggione. Quoi que de façon bien différente, et non explicitement comme nous le verrons, il nous semble que quelques-uns de leurs écrits convergent vers cette spécificité.

2.2.1 Partition exhaustive chez J.-C. Risset

Dans de nombreux écrits de J.-C. Risset, une considération récurrente est présente sur les fichiers informatiques nécessités pour la réalisation d'une pièce. Avant de traiter cette dernière, il n'est peut-être pas inutile de souligner que les propos de l'auteur sont en cohérence avec le *schème* inhérent au langage historique, Music N⁹⁴. C'est donc naturellement que ces fichiers informatiques sont nommés des partitions⁹⁵

Une considération récurrente amenée par J.-C. Risset est que ces partitions sont tantôt qualifiées "d'intégrale", tantôt "d'exhaustive". Par ordre chronologique citons par exemple :

94 cf. chapitre II 4.2.2 où ce schème est abordé.

95 Ce que M. Stroppa trouve « impropre » et qu'il préférera qualifier de « données opératoires » (p. 189)

Le programme [Music V] demande une "partition" qui lui spécifie le son voulu : cette partition est donc une recette d'obtention du son en même temps qu'une description exhaustive de sa structure physique. J.C. Risset dans [Lorrain, 1980]

Les programmes stipulant à l'ordinateur la recette de synthèse sont de véritables partitions de la structure sonore, partitions exhaustives et transmissibles [Risset, 1991]

Une partition Music 5, en même temps qu'une recette de production, est aussi une partition intégrale : elle représente l'élaboration compositionnelle de la microstructure du son [Risset, 2001]

Nous pouvons comprendre les termes d'exhaustif et d'intégral comme un prolongement du schème de Music N, soit, dans une mise en rapport de la partition de Music N (alphanumérique), et de la partition traditionnelle (graphique). Schématiquement, quand cette dernière a besoin d'un musicien pour « être rendue sonore » (nos termes), la première peut s'en passer. Mais pour ce faire, nous rappelle J.-C. Risset, le compositeur doit réaliser une « description numérique, très précise, scientifique »... qui sera ensuite « traduite en sons » [Bazolli, 2000, p. 67].

Pour nous, l'intérêt de cette réflexion se trouve non pas dans le rapport effectué entre les partitions, mais dans la relation amenée entre fichiers informatiques et phénomène sonore. Quand une approche poétique pouvait envisager ces fichiers comme des *traces*, la lecture de J.-C. Risset nous invite à les comprendre dans une relation au sonore : entre fichiers et son il y a une certaine réciprocité, de sorte que la lecture (humaine) des premiers permet une compréhension des seconds. Comme l'auteur l'introduisait déjà dans son catalogue historique, la compréhension de l'un, permet de décrire l'autre :

the computer data used for the synthesis of the sound affords a thorough description of the physical structure of these sounds. [Risset, 1969, p. 109].

2.2.2 Écriture directe chez H. Vaggione

D'une toute autre manière nous retrouvons aussi chez H. Vaggione cette relation entre fichiers et son. En prémisses il nous semble important de souligner quelques aspects du contexte avec lequel cette idée de réciprocité prend sens chez l'auteur.

Des niveaux opératoires

Pour H. Vaggione la note de la théorie musicale occidentale, à laquelle correspond un signe graphique, est une "brique syntaxique, un atome", qui appartient à un "niveau de référence" [Vaggione, 1998, p. 3a]. Et de part et autre de ce niveau se trouvent deux domaines, devenus maintenant habituels :

- le domaine du *macrotemps*, situé au-dessus de la note, "englobant toutes les échelles possibles s'étalant vers le "grand" " (*ibid.*)
- le domaine du *microtemps*, situé en dessous de la note, "englobant toutes les échelles possibles s'étalant vers le "petit" " (*ibid.*)

En considération de ces *niveaux opératoires*, qui s'ouvrent à la composition, il va de soi que H. Vaggione, et à la différence de J.-C. Risset, ne peut souscrire au schème de Music N. En effet pour le compositeur/chercheur, partition et note sur papier, ne peuvent être assimilés aux partitions et notes du langage informatique : tandis que les premiers appellent un niveau opératoire défini, une représentation de ce niveau – qui est intermédiaire au niveau macroscopique et microscopique – les seconds n'appartiennent pas à un niveau *a priori* :

la dénomination "liste de notes" est cependant trompeuse, car, bien qu'elle fasse allusion à des "listes" d'instances, les éléments de ces listes ne sauraient aucunement s'assimiler aux "notes" de la notation musicale conventionnelle – car ici les éléments des listes peuvent être de n'importe quel ordre de grandeur temporelle, et donc représenter, notamment, des durées appartenant par définition au domaine du micro temps [Vaggione, 2010, p. 64].

Objet, écriture, son

La réciprocité entre fichiers et son n'est pas quelque chose qui est *au centre* de la réflexion féconde de l'auteur. C'est quelque chose qui apparaît en creux, sans que cela soit traité substantiellement. On la retrouve par exemple dans la notion clé de l'auteur, celle d'Objet⁹⁶. Un des aspects de cette notion est qu'un Objet réunit à la fois des "faits sonores" et "des faits d'écriture" :

[la notion d'Objet] ne correspond plus à l'entité purement macroscopique définie par Schaeffer (1966), mais sous-tend désormais un autre type d'entité composée dans laquelle des structures temporelles (des process, des morphologies) se trouvent déterminées non seulement en tant que faits

96 Qu'il faut moins approcher avec Schaeffer qu'avec le paradigme de Programmation Orienté Objet des langages informatiques.

sonores mais aussi, et indissolublement, en tant que faits d'écriture [Vaggione, 1998, p. 11a].

Plus loin, dans le même texte, cette « double détermination des objets », entre sonore et écrits, amène l'auteur d'une part à qualifier ces objets de transparents, et de l'autre à parler d'une *écriture directe* :

l'objet sonore numérique est transparent, c'est-à-dire, il peut être ouvert afin d'offrir l'accès à sa structure interne, et par conséquent permettre une *écriture directe* de la matière sonore elle-même. [Vaggione, 1998, p. 13a] (nous soulignons)

De même, nous retrouvons cette idée « d'écriture directe » dans un écrit plus tardif, et cette fois-ci de façon exogène, uniquement à partir de réflexions sur le travail historique de Risset :

En fait, c'est l'étude d'une saillance morphologique [celle des sons cuivrés] qui a conduit Risset [...] à énoncer un modèle de ce caractère dynamique : un modèle opératoire, traduisible en termes paramétriques, inaugurant ainsi une "écriture directe" des phénomènes micro-temporels faisant partie de la structure physique des sons ainsi que de leur identité perceptive. [Vaggione, 2003, p. 92].

2.2.3 Écriture

Au travers de J.-C. Risset et *a fortiori* de H. Vaggione, la spécificité de la musique d'informatique nous apparaît maintenant limpide. Le son de cette musique n'est plus évanescent comme l'est celui de la musique acousmatique, mais, en reprenant les termes des chapitres précédents, il possède bien un *corrélat*. De plus, avec H. Vaggione, ce corrélat nous semble bien saisi quand il est envisagé comme une *écriture*, et peut être moins quand il est compris comme une « nouvelle causalité », ou comme un processus de production.

Avec [Zattra, 2015] qui, dans une perspective génétique, est amenée à parler *d'écriture* pour le « texte de synthèse », et surtout avec J.-C. Risset puis H. Vaggione, nous considérons que la spécificité de la musique d'informatique relève bien du fait qu'elle est écrite.

Cette spécificité peut paraître étonnante au regard de ce qui est courant d'appeler « l'écriture musicale ». Elle l'est cependant nettement moins quand on met en regard deux environnements de composition, le studio informatique et celui analogique. Si dans le second la composition se fait avec un amas de manipulations sur les appareils électroacoustiques, avec des gestes manuels, dans le premier, à la différence, au centre de l'environnement se trouvent un clavier alphanumérique et un écran.

La composition se fait alors notamment avec cette unité minimale, un symbole, dont l'écriture inévitablement découle.

L'ordinateur a été pour moi l'outil idéal parce qu'il a apporté la capacité de travailler avec des symboles discrets au niveau de la matière sonore et, en conséquence, de littéralement « écrire » des sons. H. Vaggione interviewé dans [Budón, 2013, p. 102].

3 Analyse et musique d'informatique

3.1 Positionnement

En considérant cette spécificité de la musique d'informatique nous pouvons de nouveau aborder les pratiques analytiques qui se proposent de saisir la musique « en amont du signal ».

3.1.1 La *faktura*

Comme la *faktura* peut s'appliquer à toutes les technologies de production sonore que le XXe a vu naître, elle ne nous semble pas, *de facto*, un outil analytique satisfaisant pour s'attaquer à cette spécificité de la musique d'informatique.

3.1.2 La génétique

La génétique musicale avancée dans [Zattra, 2015] semble en revanche bien traiter de cette spécificité. Comme nous l'avons vu, en son centre se trouve le concept d'écriture, duquel découle l'assimilation, très originale pour nous, des "données de synthèses" en tant que *textes*. Néanmoins, et c'est en son fondement, la génétique se situe pleinement dans la poïétique des pièces, et peu, voire pas, dans leur rendu sonore. Appliquée à la musique d'informatique, comme le dit l'auteur, l'enjeu est d'éclairer le « processus de formation d'une œuvre » (*ibid.* p. 220). Tout l'aspect sonore de ces écritures, que J.-C. Risset et H. Vaggione nous ont permis de mettre en lumière, n'est pas pris en compte. En bref, le point d'accroche de la génétique est que ces textes font partie de la genèse de l'œuvre, de son processus créateur ; s'ils permettent d'éclairer l'œuvre c'est par sa genèse.

Cet intérêt pour le processus créateur se retrouve aussi dans une critique mineure de L. Zattra vis-à-vis du travail de [Lorrain, 1981] : quand ce dernier a repris les textes de Music N de l'époque, il les a ensuite retranscrits dans un langage plus récent, Music V. Dans cette opération L. Zattra aurait aimé « une explication plus précise » des « interventions faites sur les données sources » (p. 215) – car l'on sait de nos jours que ce travail très conséquent n'a rien « de neutre », qu'il n'a rien à voir avec la copie d'un support audio⁹⁷. Dans cette même perspective relevons [Plessas, 2015] qui suggère l'emploi de logiciels de « revisionning » afin que l'on puisse avoir une trace des différentes modifications du code effectuées⁹⁸.

97 C'est ce que nous avons souhaité mettre en lumière avec le terme *design*.

98 On permettrait ainsi un travail archéologique, comme déjà le mentionnait [Stroppa, 1984, paragraphe 2].

3.2 Proposition

Nous sommes maintenant à même de finaliser la réponse à notre problématique : comment analyser les œuvres de musique d'informatique à partir de leur artefact ?

Notre premier élément de réponse est de mettre au clair la *structure de contrôle* avec laquelle la musique s'est pensée. Mais passée cette étape, une fois que les connaissances imbriquées dans l'artefact sont acquises, que l'effort de lecture des idiosyncrasies est surmonté, il reste la question « comment analyser cette musique » ?

3.2.1 Une analyse entre lecture et écoute

Notre proposition reprend en partie celle de Michael Clarke présentée dans [Clarke, 2005]. Dans celle-ci l'auteur se base sur une analyse qu'il a effectuée [Clarke, 2006] sur la pièce *Mortuos Plango, Vivo Voco* (1980) de Jonathan Harvey, pour bande 8 voies⁹⁹. Une originalité de cette publication est d'y associer un patch, réalisé dans le langage Max. En supplément (i) d'une lecture de fichiers audios qui permettent au lecteur d'entendre les extraits sonores auxquels se réfère l'auteur, et (ii) d'une carte paradigmatique aurale¹⁰⁰, le patch Max intègre des « éléments de computation » qui permettent de simuler les résultats audio de la pièce.

Cette démarche singulière est nommée par l'auteur *approche interactive aurale*. Cependant, le terme *aural* nous surprend. Il renvoie au fait que l'analyse est basée sur l'écoute (aural du latin auris («oreille») et -al) soit, en suivant notre schéma qui a ouvert ce chapitre, que l'approche analytique de M. Clarke se fait à partir du signal audio, avec l'écoute. Or, ces éléments de computation présents dans le patch en témoignent, l'analyse s'est *aussi* faite avec une saisie « en amont du signal » :

the software provides a series of interactive exercices illustrating the *techniques used* by harvey in the work. In some cases synthesis is used to simulate the *synthesis* used by the composer. *Ibid.* (nous soulignons)

Et plus explicitement :

in this analysis detailed attention has been given in particular to the *poietic* element [...]. The composer also generously made his *hand-written* sketches available and these greatly aided the process of analysis. *Ibid.* (nous soulignons).

99 Ou pour voix de garçon, son de cloche et traitements ?

100 « Aural paradigmatique chart »

Dans notre schéma de compréhension de l'analyse musicale, la démarche de M. Clarke est donc à la fois en amont et à partir du signal¹⁰¹.

3.2.2 Une analyse en amont et à partir du signal

C'est précisément dans cette réunion de pratiques analytiques que notre proposition reprend la voie amenée par M. Clarke. Le second élément de réponse à notre problématique consiste ainsi à confronter ces connaissances acquises suites à la mise au clair de la structure de contrôle, à l'écoute de la pièce, sa perception. S'aider de l'appropriation de l'artefact qui a été réalisé pour écouter la pièce. Ou bien, réciproquement, l'analyse doit maintenant expliquer ce qu'il écoute en répondant à la question « comment cela est fait ? ». Il s'agit donc d'expliquer comment la composition est faite *dans le temps* de la pièce.

En confrontant la structure de contrôle à l'écoute de la pièce, et inversement, une prémisse de notre proposition est donc de retrouver une circulation de significations entre ce qui est lu et ce qui est perçu.

Ainsi, là où nous avons l'habitude d'avoir uniquement un feedback auditif, nous avons maintenant une confluence de l'oreille et des yeux ainsi que la possibilité de stocker nos actions sous forme de codes. Je pense que c'est un aspect très positif de la musique informatique. H. Vaggione interviewé dans [Budón, 2013, p. III6].

C'est bien cette circularité qui apparaît singulière à cette musique d'informatique.

¹⁰¹ Il faut préciser que la démarche de M. Clarke déborde des pièces d'informatique musicale. En effet dès [Clarke, 2005] l'auteur voit des possibilités d'application aux musiques de tradition aurale. Ensuite, dans [Clarke, 2009], titré « Extending Interactive Aural Analysis : Acousmatic Music », l'auteur applique et renouvelle son approche avec la pièce *Wind Chimes* (1987) de Denis Smalley, où la composition, par comparaison à celle de *Mortuos Plango, Vivos Voco* (J. Harvey), s'est faite de façon « intuitive bien plus que analytique » (*Ibid.*). Enfin, en 2012, le positionnement de l'auteur vis-à-vis de son approche est plus clair : « Nonetheless, it would be unfortunate to [...] restrict the interactive aural approach to the electroacoustic repertoire », et plus loin « In principle, the interactive aural analysis approach can be applied to any piece, though each new work will inevitably lead to extension and adaptation of the approach » [Clarke, 2012]. L'approche de l'auteur s'est donc développée à partir des musiques d'informatique pour ensuite s'ouvrir très largement à « n'importe quelle pièce ».

Troisième partie
Application

Jupiter (...) includes the "first use" of more technologies than you can shake a stick at.

Andrew May

Les machines accélérant le temps, elles posent beaucoup de problèmes, mais nous les aimons.

Philippe Manoury

Chapitre VI

Introduction à l'analyse de *Jupiter*

Résumé

Ce chapitre est une introduction à l'analyse de la pièce *Jupiter* de Philippe Manoury. Dans une première partie nous introduisons la pièce – nous décrivons globalement sa riche histoire technologique –, et dans une deuxième nous donnons une vue globale de l'artefact, entre système de synchronisation, Modules, Algorithmes et Routing. Enfin dans la dernière partie nous présentons brièvement la structuration de la pièce.

1 Introduction à l'analyse

1.1 *Jupiter*

Jupiter est une pièce mixte commandée par l'Ircam pour flûte traversière et électronique temps réel. Elle est créée le 28 avril 1987, dans l'espace de projection, avec un programme qui célèbre les 10 ans de l'institut. Elle est alors interprétée à la flûte par Pierre-André Valade, et l'électronique est gérée par Miller Puckette¹⁰², Marc Battier en est l'assistant musical. La pièce est dédiée à la mémoire de Larry Beauregard, flûtiste de l'Ensemble Intercontemporain, disparu en 1985¹⁰³, qui fut à la genèse de la pièce.

Jupiter fait date dans la musique contemporaine pour les liens qui sont tissés entre un instrumentiste et une production électronique. En effet, c'est bien au centre de *Jupiter* que sont placées les interactions entre deux « univers sonores », comme l'illustre l'effectif :

Jupiter fut composé avec la contrainte de n'avoir pas d'autres interlocuteurs que la seule 4X. J'ai volontairement éliminé tout recours à un quelconque ensemble instrumental voulant développer au maximum les rapports entre le soliste et la machine [Manoury *et al.*, 2003, la flûte 4X]¹⁰⁴

On peut relever à cette même époque d'autres pièces qui explorent les interactions possibles entre instrumentistes et électronique, et ce en dehors d'une « musique sur bande ». Citons par exemple *Alone* (1987) de Thierry Aloni, pour mezzo-soprano, chœur d'enfants, ensemble et électronique, qui fait aussi appel à la 4X et à la flûte MIDI, et qui de même est créée pour les 10 ans de l'Ircam ; ou encore *Diadèmes* (1986) de Marc-André Dalbavie, pour alto, ensemble instrumental et électronique. Néanmoins, *Jupiter* se démarque car l'interaction réalisée va « un pas en avant », dans le sens où elle intègre, en plus d'une possibilité de transformation d'un instrument – ce qui est présent dès *Mantra* (1969-70) de Stockhausen –, d'une part un suiveur de partition, et de l'autre des algorithmes qui vont lier l'interprétation instrumentale à la production électronique – ce qui donnera plus tard naissance au concept de *partition virtuelle*, cher au compositeur [Manoury, 2007].

De cette possibilité de composer, grâce à l'outil l'informatique, des interactions entre instrumentistes et électronique, naît, chez le compositeur, et par contraste avec une musique sur

¹⁰² Sur le programme on peut lire « contrôle 4X ».

¹⁰³ Relevons que *Memoriale* (1985) de Pierre Boulez est aussi dédié à la mémoire de ce jeune flûtiste.

¹⁰⁴ Cette référence renvoie à un DVD multimédia qui regroupe différents textes. Afin que le lecteur puisse retrouver le texte dont il est question, nous indiquons, après la date d'édition, le titre du texte. Cette méthode est employée tout au long de notre écrit.

bande, la musique interactive, ou encore la musique du temps réel (cf. [Manoury, 2007]). Afin de développer cette nouvelle possibilité d'interaction le compositeur y consacre un cycle de quatre pièces, au titre évocateur *Sonus Ex Machina*. À *Jupiter*, pièce pionnière donc, succèdent *Pluton* (1988), pour piano et électronique, puis *La partition du ciel et de l'enfer* (1989), pour flûte midi, piano, piano midi et ensemble et électronique, et enfin *Neptune* (1991), pour trois percussionnistes et électronique.

1.2 Histoire et technologie

Jupiter a un passé technologique très conséquent. Pour reprendre Peter Szendy, c'est une pièce qui hérite d'une « culture du concert », où « l'interaction entre *interprètes* et la « machine » est privilégiée » (cf. [Szendy, 1998]). Par conséquent, comme elle date de la période pionnière et qu'elle est toujours actualisable, elle a inévitablement dû faire face aux nombreuses obsolescences technologiques des machines qu'elle a sollicitées :

[Jupiter] détient sans doute le record du nombre de plateformes d'informatique musicales sur lesquelles elle est passée (...) : au moins cinq systèmes différents (4X, ISPW, JMax, MaxMsp, PureData) [Lemouton, 2012, p. 18].

Symboliquement, la machine avec laquelle elle a été créée, la célèbre machine 4X, est depuis 2009 une des pièces de la collection du Musée de la musique de la Philharmonie de Paris. L'histoire technologique de *Jupiter* peut alors se dessiner entre ces deux dates, de l'usage pionnier de la 4X à son intégration dans un musée. Rendre compte de cette histoire, et nécessairement des différentes « versions » de la pièce impliquées lors de changements de machine, nécessiterait, à l'instar des travaux génétiques mené par Laura Zattra (cf. chapitre VI), un travail à part entière qui ne peut s'inscrire dans notre démarche. Nous avons donc choisi d'éclairer cette riche histoire au travers de quatre « moments » : le premier concerne bien évidemment la création ; le deuxième un portage vers la machine ISPW ; le troisième l'ajout d'un module de synthèse (PAF) ; et le dernier, plus flou dans la datation, concerne le portage vers les langages Max et Pure Data, devenus maintenant standards.

1.2.1 La création avec la 4X

Comme en témoigne le compositeur, à l'époque de la création (1987) la technologie déployée est véritablement colossale : « Il ne fallait pas moins de quatre ordinateurs différents pour réaliser toutes les tâches nécessaires à la réalisation de cette œuvre ! » [Manoury *et al.*, 2003, *Jupiter*]. Si les moyens technologiques sont si conséquents, c'est que ce que réalise la pièce est pour le moins novateur. En effet, comme nous l'avons vu, *Jupiter* est une des premières pièces à

utiliser des technologies de synthèse et de traitement sonore synchronisées à une interprétation instrumentale. Mais, plus justement, ce qui est notable c'est que ces technologies du temps réel ne sont pas encore stabilisées : *Jupiter* est concomitant aux avancées technologiques du temps réel, elle est composée avec des ébullitions technologiques :

Lorsqu'en 1986, j'entrepris la composition de *Jupiter*, je n'avais guère de modèle à ma disposition. Les systèmes d'interaction en temps réel entre instrument traditionnel et un synthétiseur étaient encore balbutiants [Manoury, 1996]

La production audio en direct est réalisée par la machine 4X¹⁰⁵, inventée par Giuseppe di Giugno au début des années 1980 (cf. [Giugno *et al.*, 1981]). Pour communiquer avec cette dernière, Miller Puckette conçoit un langage de programmation textuel, dédié au contrôle, nommé, en hommage au concepteur de Music N, Max (cf. [Puckette, 2002a])¹⁰⁶. La synchronisation homme machine est alors réalisée par l'intermédiaire d'une « flûte-4x ». Réalisée par Lawrence Beauregard et Xavier Chabot en 1982, il s'agissait d'une flûte augmentée de 15 capteurs mécaniques, permettant de transmettre à l'ordinateur le doigté du flûtiste.

À l'aide de cette représentation informatique du doigté du flûtiste, et d'une copie de la partition, la machine 4X était alors capable de suivre le jeu du flûtiste.

Enfin, relevons que pour cette première version, *Jupiter* possédait une section supplémentaire, soit 14 sections et non 13 comme nous le verrons dans ce chapitre (cf. [Derrien *et al.*, 1995, p. 44]).

105 Qui était, en 1984, "the most powerful digital synthesizer/audio processor in the world" [Puckette, 2002a, p. 4]

106 Même s'il n'a pas encore son apparence graphique, ce langage peut être vu comme l'ancêtre des langages maintenant très connus Max et Pure Data. Avant Max, M. Puckette travaillait sur la conception d'un autre langage, Music500, qui n'a pas eu le temps d'être effectif. Ce dernier était directement inspiré d'un autre langage, RTSKED, conçu par Max Mathews. cf. [Puckette, 1984] et [Puckette, 2002a]



*Fig. 14 - la machine 4X et la flûte 4X, vue du musée
© Musée de la musique – Philharmonie de Paris*

1.2.2 L'Ircam Signal Processing Workstation (ISPW)

Le deuxième moment a lieu en 1992 lors d'un portage vers une nouvelle machine, l'ISPW (Ircam Sound Processing Workstation) – un ordinateur Next équipé de 3 cartes DSP personnalisées –, inventée par l'équipe d'Eric Lindemann (cf. [Lindemann *et al.*, 1991]). La production audionumérique se fait avec un langage propre à la machine, nommé Max/FTS. Ce dernier, inventé par Miller Puckette (cf. [Puckette, 1991]), est maintenant équipé de la « face graphique » (GUI¹⁰⁷) si caractéristique de ces langages. Les technologies ayant avancé, l'appareillage mécanique de la « flûte-4X » est alors remplacé par l'analyse en temps réel du

¹⁰⁷ Graphic User Interface

signal audio : la note jouée par la flûte n'est plus déduite des positions des doigts¹⁰⁸, mais du son capté... comme le ferait un musicien qui suit une partition.

1.2.3 L'ajout des PAF

Le troisième moment date de 1996, il est nettement plus particulier puisqu'il consiste en l'ajout d'un module de synthèse, nommé PAF (Phase Aligned Formant), qui sera abordé *in extenso* au chapitre suivant. En substance, précisons simplement ici que cette méthode est inventée par M. Puckette au début des années 1990. Philippe Manoury l'intégrera alors dans deux de ses pièces du cycle *Sonus ex machine*, *Jupiter* et *La partition du ciel et de l'enfer* (1988) (cf. [Puckette, 1996]).

L'intégration de ce module de synthèse dans *Jupiter* s'est fait avec le chercheur et compositeur Tom Mays. Par la suite P. Manoury en a profité pour revoir certains Modules de synthèse, par exemple des samples de flûte ont été ajoutés. C'est à l'issue de ce travail que la version définitive fut fixée :

Ce n'est qu'au cours du mois de novembre 1996 que je décidais d'établir la version définitive en modifiant de façon substantielle le contenu des programmes de synthèse, opération qui peut s'apparenter à une réorchestration [Manoury, 1997].

1.2.4 Vers Max et Pure Data

Enfin, le dernier moment correspond au portage de la version ISPW vers des langages maintenant plus indépendants d'un hardware, comme Max, jMax et Pure Data. Dans la base Sidney de l'Ircam¹⁰⁹ la référence la plus ancienne d'un tel patch date de 2002. Il s'agit d'un patch Max/MSP pour un ordinateur Macintosh G4 sous OS9. Parallèlement, en 2003 M. Puckette effectue un portage vers Pure Data, pour un système Linux.

108 Ceci reste néanmoins une réduction puisque pour la flûte-4X le signal audio était aussi capté. Il permettait de savoir si c'était le fondamental ou l'harmonique de rang 1 qui était joué avec tel doigté (cf. [Manoury, *et al.*, 2003, la flûte 4X])

109 Sidney est la base de données de l'Ircam qui rassemble les documentations et les programmes informatiques des pièces avec électronique du répertoire cf. [Lemouton, 2016]. Cette base est accessible en ligne avec un compte utilisateur (<http://brahms.ircam.fr/sidney/>, 01 04 2017). Voir aussi l'enregistrement de l'intervention de Serge Lemouton du 6 Juin 2017, « Composer pour la postérité à l'époque de l'obsolescence programmée », dans le cadre du colloque *Etat de l'art / Etat d'alerte*, organisé par P. Manoury au Collège de France (<http://www.college-de-france.fr/site/philippe-manoury/symposium-2017-06-06-06-09h00.htm>, 10 07 2017).

1.3 Matériau pour l'analyse

Pour mener notre analyse nous avons eu recours à différents matériaux. En premier, l'archive informatique qui permet d'actualiser la pièce, qui contient l'usuel patch, en deuxième la partition, et enfin inévitablement un enregistrement de la pièce.

1.3.1 Les patches

Dans notre démarche le matériau que nous avons le plus utilisé est ce qui est communément appelé le patch. Nous en avons utilisé deux. Le premier, celui sur lequel nos travaux se sont basés, est le patch Pure Data. Il résulte donc du portage effectué par M. Puckette en 2003. Il a été mis à disposition dans le cadre d'un projet, nommé *Pure Data Repertory Project*, aussi novateur qu'ambitieux, dont le but était de partager les artefacts des pièces standards associant instrumentistes et électronique (cf. [Puckette, 2001])¹¹⁰.

Le second est un patch pour Max daté de 2011. Nous l'avons obtenu grâce à l'Ircam que nous remercions ici. Il est disponible sur la plateforme Sidney de l'institut. La datation de ce dernier est plus facile puisqu'il correspond à une présentation de la pièce, faite à Dijon le 22 octobre de l'année 2011.

Pour aller de pair avec ces patches nous avons eu besoin d'un enregistrement de la flûte solo. Il est présent dans l'archive Max, c'est un fichier nommé *jupiter-sim-markers.aiff*, qui correspond à l'enregistrement d'une interprétation réalisée par Sophie Cherrier.

1.3.2 Partitions

Nous avons utilisé deux partitions. La première est manuscrite et date de la création ; la seconde est gravée, éditée par les éditions Amphion et datée de 1991. Sans faire une comparaison complète entre ces partitions – ce qui resterait néanmoins fort intéressant – nous pouvons constater que la partition gravée contient nettement plus d'information que celle manuscrite¹¹¹.

1.3.3 Enregistrement

Enfin notre écoute s'est faite avec une version enregistrée de 1997, interprétée par Sophie Cherrier¹¹² (Paris, Adès, 1997). En plus de cet enregistrement, nous avons eu recours à de nombreuses simulations avec le patch Pure Data et le fichier audio de flûte solo. Ces « écoutes

¹¹⁰ En plus de *Jupiter* et *Pluton* sont inclus *13 loops* (1988) de Rand Steiger, et *Mantra* (1970) de Stockhausen.

¹¹¹ Par exemple elle inclut des notations pour la PAF et pour la synchronisation du Chapo, ce qui est absent de la partition manuscrite.

¹¹² Il existe aussi une version interprétée par Elizabeth McNutt en 2000. Cd « pipe wrench: flute + computer » (New York, Electronic Music Foundation, DOI [10.1162/comj.2001.25.3.101](https://doi.org/10.1162/comj.2001.25.3.101)). C'est cette version qui est utilisée dans l'analyse de [May, 2006].

de laboratoire » nous ont par exemple permis d'écouter les Modules séparément, de les isoler de « l'orchestre numérique » (expression empruntée à [Bonardi, 2015]), ce qui nous a été essentiel pour la saisie des Modules, et *a fortiori* pour ceux de traitement qui sont souvent employés en groupe comme nous le verrons.

1.4 Outil pour l'analyse

Dans notre travail de compréhension de l'artefact nous avons eu recours à de maintes reprises à un langage de programmation *ad hoc*, nommé Processing. Il nous a permis de réaliser différents programmes qui nous ont fortement aidés pour naviguer dans la densité de l'artefact. En suivant la vue de la boîte noire on s'aperçoit que tous nos programmes respectent le même schéma : ils chargent en entrée la partition électronique (les fichiers qlist), et en sortie ils structurent et/ou représentent des « informations de plus haut niveau ».

La réalisation de nos programmes peut se comprendre comme la formalisation de représentations, que nous avons acquises après la lecture de l'implémentation des Modules. À partir de ces représentations, nos programmes effectuent des lectures des fichiers qlist afin d'en extraire des significations musicales.

Nos programmes ont permis de réaliser quatre travaux fondamentaux pour notre analyse :

1. la fouille et la structuration de toutes les variables présentes dans les fichiers qlist ;
2. l'extraction des écritures pour chacun des Modules ;
3. l'implémentation de règles pour déduire les chaînes du routing ;
4. l'implémentation de règles pour déduire les états des Modules ;

Nous détaillons ces programmes dans le chapitre VIII.

2 Vue globale de l'artefact

Comme en témoigne la phrase d'Andrew May placée en exergue, l'artefact de *Jupiter* est très conséquent. Avant d'amener des considérations sur le cœur de l'artefact – à savoir les trois éléments clés que nous avons définis au chapitre V – il nous est nécessaire d'aborder le dispositif de spatialisation et le système de synchronisation.

2.1 Diffusion électroacoustique

En plus de la flûte, 7 haut-parleurs sont requis : un retour pour le flûtiste, une diffusion stéréophonique et un système quadriphonique qui entoure le public.

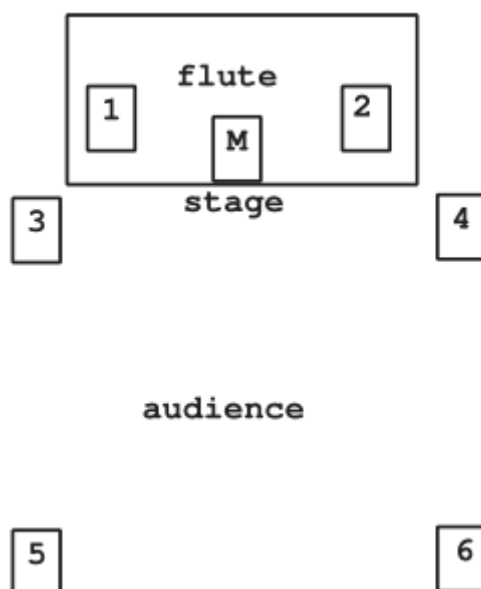


Fig. 15 - La disposition des Haut-Parleurs

Dans le patch les diffusions stéréophonique et quadriphonique sont respectivement nommées *stereo* et *quadra* – termes que nous reprenons par la suite.

2.2 Le système de synchronisation

Cette technologie permet de synchroniser la partition électronique, prévue dans le temps de la composition, au jeu de l'instrumentiste, qui se déploie dans le temps de l'interprétation. Elle

permet ainsi de réaliser une des idées phares de la musique du temps réel : l'électronique peut se concevoir en interaction avec un instrumentiste.

On peut comprendre ce système comme un ensemble de moyens technologiques développé pour permettre à l'électronique de se déployer *en dehors* d'un temps figé, inscrit sur un support¹¹³. Dans la mesure où c'est à partir de ce système que la composition peut se réaliser, soit à partir d'une interaction entre instrumentiste et électronique, nous pouvons le comprendre comme un soubassement de l'artefact.

Puisque la machine reconnaît tout ce que joue l'instrumentiste il est possible de synchroniser n'importe quelle action programmée avec n'importe quel événement musical. Le tout est affaire de goût et de composition proprement dite [Manoury, 2003, flûte-4X].

De cette volonté de synchroniser l'écriture électronique à celle instrumentale, nous pouvons en déduire l'importance temporelle des algorithmes : ils ne seront pas en lien avec la macrostructure de la pièce mais à des niveaux nécessairement infra. L'inscription temporelle des algorithmes devrait donc être d'une durée inférieure à celle des sections de la pièce (~ 4 min).

Nous détaillons ci-après le fonctionnement de ce système clé de synchronisation. Il se compose de trois éléments : (i) une copie numérique de la partition de la flûte, (ii) la partition électronique, et (iii) des événements qui permettent de synchroniser les deux dernières « partitions ».

Au niveau de l'implémentation, le système fonctionne avec trois objets de programmation clés, maintenant bien connus des spécialistes. Le premier permet l'analyse d'un signal audio, [fiddle~]¹¹⁴, le second permet de suivre le jeu instrumental, [scofo]¹¹⁵, et le dernier, plus simple, permet la lecture de la partition électronique, [qlist]. Tous ces objets ont été développés dans le langage C par M. Puckette. La technologie nécessitée pour réaliser ce système est donc très conséquente. Par exemple l'objet [scofo] est fait de 800 lignes de code en C. Dès lors, notre compréhension se fait en adoptant le principe de la boîte noire : il ne s'agit pas d'éclairer de quoi est fait le système – soit ici des fichiers d'implémentation – mais ce que fait le système.

113 Pour une vue historique sur le suiveur de partition voir [Count, 2008, p. 125].

114 Cette forme empruntée au manuel Pure Data [Puckette, 2005] permet d'indiquer les noms des objets du langage. Il s'agit donc du vocabulaire du langage de programmation.

115 [scofo] n'est pas inclut nativement dans le langage Pure Data ; il est donc présent dans le dossier *lib* de l'archive.

2.2.1 Des événements

La synchronisation instrument/électronique se fait avec une discrétisation forte, avec des « points de synchronisation », plus communément appelés des *événements*¹¹⁶ (cue). Ils permettent de faire correspondre des actions électroniques à une note précise présente dans la partition de la flûte. Ces événements ne sont pas présents à chaque note : c'est le compositeur qui choisit les instants où l'électronique doit se synchroniser. Parallèlement, les actions à réaliser sont définies dans la partition électronique sous forme alphanumérique comme nous le verrons.

2.2.2 Copie de la partition instrumentale

Pour réaliser la synchronisation une copie informatique de la partition de flûte est présente dans l'artefact. Elle est chargée dès l'ouverture du patch avec l'objet [scofo]. Cette copie numérique est faite de 13 fichiers textes, une part section, tous placés dans le dossier *score* de l'archive. Leur syntaxe est la suivante : durée (ms), valeur de pitch (MIDI), et numéro d'événements. Voici un extrait de cette copie numérique pour les événements 31 à 34 de la première section, suivi de l'équivalent dans la partition manuscrite.

12690	77	31
975	83	32
917	82	33
1057	74	34
313	66	

Fig. 16 - Extrait de la partition manuscrite, Section 1

2.2.3 Fonctionnement

Le signal de la flûte est adressé au patch. Il est analysé en hauteurs MIDI à l'aide de l'algorithme d'analyse [fiddle~] (cf. [Puckette *et al.*, 1998]). Ce dernier permet donc de passer

¹¹⁶ Pour alléger la lecture on utilisera parfois l'abréviation *evt*

du signal audio à des valeurs de notes MIDI, soit, du domaine du signal à celui du symbole. Ensuite, un léger calcul temporel est implémenté. Il permet de fournir la durée (en ms) pendant laquelle « la note »¹¹⁷ n'a pas changé. C'est cette liste de deux données (pitch, time) qui est envoyée dans le système de synchronisation [scofo].

On devine maintenant la fonction de [scofo] : à l'aide (i) de ce flux de données (pitch, time) et (ii) de la copie numérique de la partition instrumentale, l'objet va déduire la position du flutiste, et par conséquent lire les événements indiqués par le compositeur.

2.2.4 La partition électronique

Comme la copie numérique, la partition électronique est faite de 13 fichiers textes, tous chargés dès l'ouverture du patch via l'objet [qlist], et tous placés dans le dossier *score*. Afin d'alléger nous utilisons le terme *qlist* pour désigner tous ces fichiers. À titre d'exemple voici le contenu d'un de ces fichiers pour l'événement 32 de la première section.

```
0 32 ----- 410;
amp3 0;
trans0 500;
125 trans1 1000;
125 trans2 1100;
hamp1 0;
125 amp3 110 5;
rgate 117;
revfb 127;
trans3 1900;
hamp2 0;
125 amp4 110 5;
trans0 1700;
hamp3 0;
125 rgate 0;
trans1 1000;
hamp4 0;
125 trans2 500;
hamp1 0;
125 trans3 1100;
hamp2 0;
```

Cette partition électronique se structure donc avec les événements. Leurs valeurs sont inscrites après un *o*, et avant la suite de caractère -----¹¹⁸ ce qui facilite grandement la lecture. La synchronisation n'est cependant pas aussi brutale puisque les actions peuvent être séquencées à l'intérieur d'un même événement, en précisant des temps de délai. C'est ce qui est présent dans ce dernier exemple. Nous pouvons ainsi résumer la lecture des qlist :

¹¹⁷ Nous ajoutons des guillemets car il y a nécessairement du bruit en sortie de l'algorithme d'analyse.

¹¹⁸ Qui s'avère être une variable !

- si la ligne ne commence pas par un nombre, alors l'action est envoyée et la ligne suivante est lue ;
- si la ligne commence par un nombre, alors ce nombre est un temps de délai à appliquer avant d'envoyer l'action ;
- si la ligne annonce un nouvel événement, alors la lecture est interrompue.

Par conséquent nous voyons que les temps de retard sont *cumulatifs* : l'action présente à la 9^e ligne (*trans3 400*) est envoyée 375 ms après l'événement 32.

2.2.5 Représentation

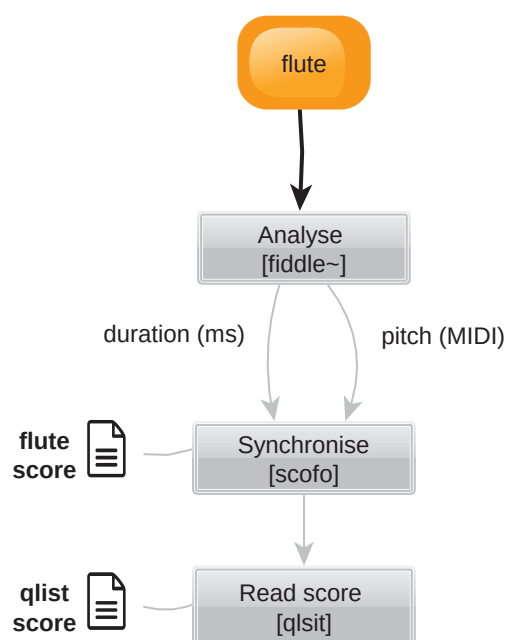


Fig. 17 - Le système de synchronisation

Malgré cet arsenal technologique il faut préciser que la synchronisation ne se fait pas intégralement par la machine. Il arrive en effet pour certains passages que la synchronisation se fasse par un Homme en régie. Ce dernier suit le jeu du flûtiste et indique à la machine l'arrivée des nouveaux événements. On retrouve ainsi dans la fenêtre principale du patch un important « bouton », nommé *step*, qui permet de faire passer le système de synchronisation à l'événement suivant.

2.3 Modules et Algorithmes

2.3.1 Modules

L'artefact se dessine avec 9 Modules, répartis quasi équitablement selon la typologie synthèse et traitement : 4 de synthèse et 5 de traitement. Les quatre Modules de synthèse sont nommés Sampler, Additive, Chapo et Paf, et ceux de traitement, sont Harmonizer, Frequency Shifter, Reverb, Noise, et Spat. Tous sont décrit au prochain chapitre.

Dans [May, 2006] on peut trouver un autre module nommé « Phase Modulation ». Ce dernier est bien présent dans le patch mais nous ne le considérons pas comme un Module de l'artefact, pour la raison qu'il s'avère doublement figé. Une première fois au niveau du routing interne : alors que tous les Modules de traitement sont reliés au routing interne, celui-ci ne l'est pas, il est appliqué sur le signal de flûte tout le long de la pièce. Une deuxième fois au niveau de son écriture : tout le long de la pièce ces paramètres reçoivent les mêmes valeurs. Conséquemment, ce traitement, qui permet d'enrichir le signal de la flûte, « de créer une sorte de flou (...) comme il s'en produit dans un chœur » [Manour *et al.*, 2003, *Jupiter*], appartient au domaine du mixage. On pourrait le comprendre comme un traitement ajouté par un ingénieur du son sur les signaux produits par l'artefact... soit non comme un Module avec lequel se fait la composition¹¹⁹.

2.3.2 Algorithmes

Deux Algorithmes correspondant à des idées musicales sont présents dans l'artefact. Bien que ceux-ci peuvent engendrer des sonorités différentes, ils restent néanmoins attachés à des Modules spécifiques. Dès lors, après une saisie entière de l'artefact au prochain chapitre, et comme nous l'avons vu au précédemment de façon théorique, nous considérons ces deux algorithmes comme faisant partie intégrante de Modules, en l'occurrence du Sampler et du Chapo.

2.3.2.a Les interpolations du Sampler

L'Algorithme d'interpolation permet de créer de nouvelles séquences, de l'ordre de la minute, en transgressant la théorie occidentale à la fois sur le plan des hauteurs et sur celui du rythme. Au niveau des hauteurs, une nouvelle carte de fréquences est inventée à partir de la gamme chromatique ; au niveau du rythme des interpolations successives entre deux motifs rythmiques sont établies, de sorte que le résultat obtenu échappe à la notation occidentale.

Au-delà des calculs effectués, ce dernier algorithme reste particulier dans la mesure où il vient marquer une fissure entre le poétique et l'esthétique. Dans sa réalisation, il y a en effet une

¹¹⁹ Le lecteur relèvera la distinction entre module de synthèse/traitement et Module de l'artefact.

volonté de tisser un lien entre la flûte et l'électronique dans le temps du concert (musique interactive). L'interpolation rythmique se fait ainsi à partir d'enregistrements en direct de motifs joués à la flûte – donc variables d'une interprétation à l'autre. Cependant, la distance temporelle entre ces enregistrements de la flûte et les séquences calculées par l'algorithme est telle (plus de 3 minutes) qu'il reste impossible à l'écoute de lier ces derniers. De plus, après une saisie de l'artefact, et par corrélation, nous nous sommes aperçus que l'artefact ne contenait pas tous les calculs nécessités par un tel algorithme : seuls des calculs rythmiques sont en effet présents. Avec notre implémentation nous nous sommes ainsi aperçus que les séquences du Sampler n'étaient pas calculées dans le temps du concert mais lues à partir de fichier. Nous détaillons ce point plus amplement au prochain chapitre.

2.3.2.b Chapo

Le Chapo correspond lui pleinement aux idées d'une musique interactive : il établit une interaction, de l'ordre de 20 ms, entre la flûte et un son électronique. D'abord, le son électronique est une sommation d'une vingtaine de sinusoïdes, prescrites dans le temps de la pièce, et sur ces sinusoïdes est apposée une enveloppe spectrale faite de 5 formants. Ensuite, cette enveloppe spectrale se déplace vers le grave ou vers l'aigu en fonction des mouvements en ambitus que la flûte est en train de jouer. À l'écoute nous pouvons ainsi aisément faire un lien entre flûte et électronique, d'autant plus que l'écriture de la flûte contient parfois des intervalles de plus d'une octave, ce qui engendre des mouvements électroniques très limpides.

2.3.3 Les Modules dans l'implémentation

Au niveau de l'implémentation, de l'apparence de l'artefact, le nombre de Modules est amené à changer. D'abord la flûte est présente à l'instar d'un module : on peut acheminer son signal vers les Modules de traitement. Ensuite les Modules Additive, Chapo et Paf sont regroupés dans une même entité, nommée *Osc*. Ce regroupement vise simplement à réduire le nombre de variables à créer. Par exemple, au lieu d'avoir trois contrôles d'amplitude pour l'envoi vers la diffusion, il n'y en a ici plus qu'un. De même, pour le routing interne, ce regroupement permet une économie considérable de variables.

Tous les modules possèdent une lettre d'abréviation qui est ensuite réutilisée dans les noms des variables. Le tableau ci-dessous regroupe, entre autres, les Modules et leurs abréviations.

Module	Type	Grou- pement	Algorithme	Nb signaux	Abbré - viation
Flute	Ø			1	d
Sampler	Synth		oui	4	t
Additive	Synth	osc		4	o
Chapo	Synth	osc	oui	4	o
Paf	Synth	osc		1	o
Harmonizer	Treat			1	h
Frequency Shifter	Treat			2	f
Reverb	Treat			2	r
Noise	Treat			1	n
Spat	Ø			4	4

Tab. 3 - Vue synthétique sur les Modules de l'artefact

2.3.4 Représentation des Modules et du système de synchronisation

L'ensemble, entre analyse de la flûte, système de synchronisation, Modules et algorithmes peut ainsi être représenté :

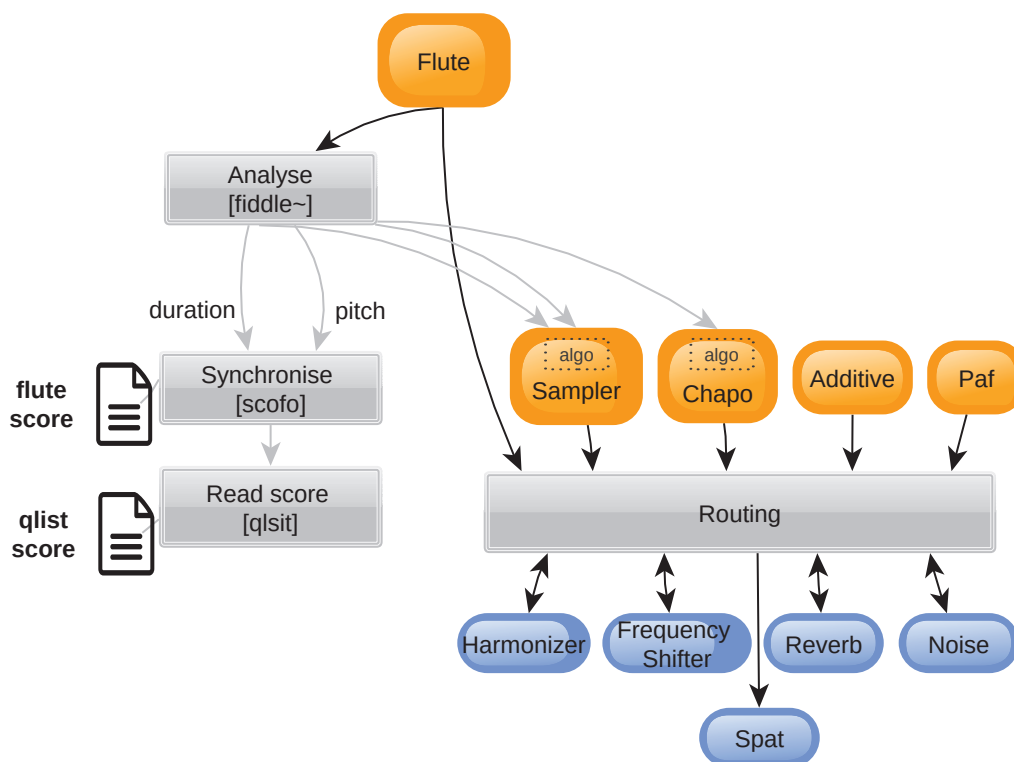
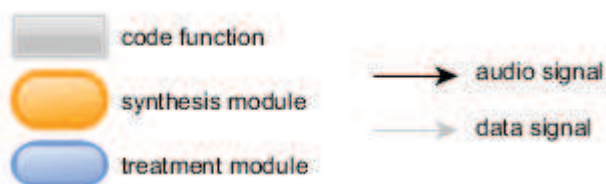


Fig. 18 - Le système de synchronisation et les Modules



On relève la spécificité du Spat, qui s'apparente à un module de traitement, mais qui ne peut évidemment pas redistribuer son signal dans le routing interne. Le Spat, à la différence de tous, ne peut se penser sans l'implantation spatiale du dispositif de diffusion. Ainsi, quand nous parlons des Modules de traitement, le Spat est exclu.

2.4 Le routing

2.4.1 Le routing interne

Le routing est le dernier élément clé de l'artefact. Il est particulier puisque, comme nous l'avons vu, il autorise un *routing interne* entre Modules. À chaque événement le compositeur a donc la possibilité de reconfigurer le routing, soit par exemple d'acheminer le signal de flûte vers l'Harmonizer, puis vers la Reverb, puis, à l'événement suivant, de supprimer cette chaîne et d'en créer une nouvelle de la flûte vers le Frequency Shifter.

L'implémentation réalisée permet une très grande flexibilité. D'abord tous les Modules possèdent une sortie monophonique, qui est obtenue en effectuant la sommation de toutes les voies¹²⁰. Ensuite, tous les Modules de synthèse sont reliés à ceux de traitement, et tous ceux de traitement sont reliés entre eux.

Le son de la flûte est reconnu et envoyé instantanément dans différents modules permettant soit de le maintenir dans le temps (réverbération prolongée aussi longtemps qu'on le désire) (pointer vers traitements électroniques), soit de le transporter dans l'espace (harmoniques modifiant la hauteur sans altérer la durée) pour en former des configurations harmonico-polyphoniques, soit, enfin, de transformer son timbre (frequency-shifters) [Manoury, 1987]

La figure suivante présente l'implémentation sous Pure Data du routing interne, elle témoigne notamment des nombreuses interconnexions possibles entre Modules.

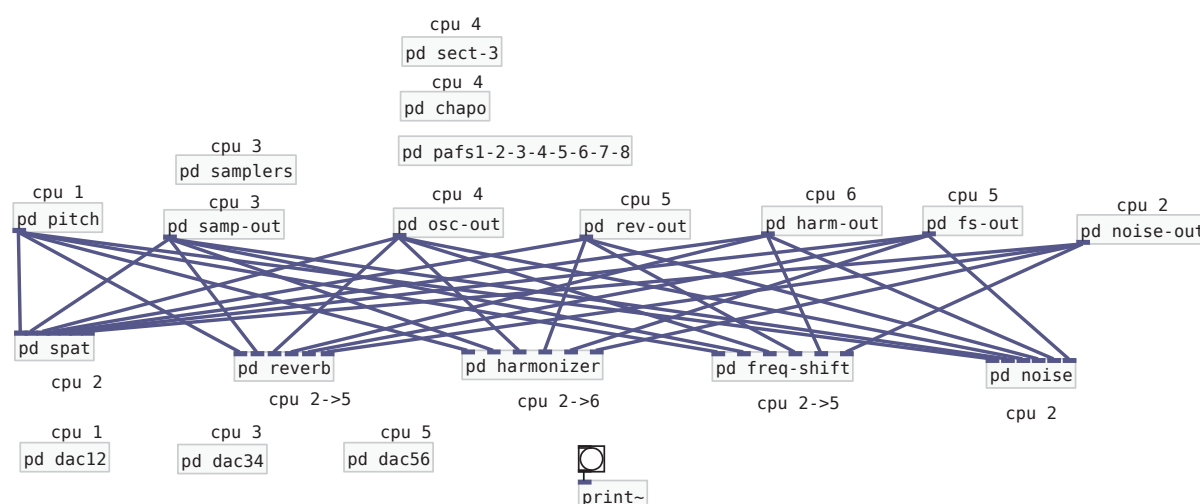


Fig. 19 - Implémentation du routing interne dans le patch

¹²⁰ Seule la Reverb n'obéit pas à cette règle comme nous le verrons au chapitre suivant.

Les Modules de traitement sont situés dans le bas du patch. Chacun d'eux possède 6 entrées, soit une pour chaque Module (le signal de la flûte est présent dans l'objet nommé *pitch*; Additive, Chapo et Paf sont regroupés dans *osc-out*).

Le paramétrage du routing se fait avec des variables qui contrôlent l'amplitude du signal pour chacune de ces connexions. Comme il y a 6 modules et 5 destinations possibles, la configuration complète du routing nécessite pas moins de 30 variables. Le nom de ces dernières est simplement fixé selon une règle qui utilise les abréviations des Modules : *ttof* désigne la connexion du *Sampler* vers le *Frequency Shifter*, *htor* la connexion de l'*Harmonizer* vers la *Reverb*, etc.

Cette flexibilité remarquable entraîne néanmoins deux contrecoûts : des boucles à l'infini (larsen) peuvent être créées, et d'autre part, comme une configuration complète du routing nécessite 30 lignes dans la partition électronique, l'écriture du routing engendre une certaine lourdeur, soit, plus fortement, comme nous le verrons, une *difficulté de représentation*.

2.4.2 Le routing externe

Il est évidemment plus simple. Tous les Modules sont reliés à la diffusion et le compositeur a non seulement la possibilité de doser la quantité de signal à envoyer, mais aussi de supprimer des connexions. Cette possibilité s'avère importante pour les Modules de traitement. Il est en effet question de doser leur importance, à savoir si le signal traité (*wet*) doit être diffusé seul ou en parallèle de celui non traité (*dry*).

Stereo	Quadra
Frequency Shifter	Sampler
Reverb	Additive
	Chapo
	Paf
	Harmonizer
	Noise
	Spat

Tab. 4 - Le routing externe

2.4.3 Routing interne : difficulté de représentation

Puisque le routing interne se contrôle avec une trentaine de variables, une configuration entière nécessiterait le même nombre de lignes dans les qlist. On comprend alors que l'écriture du routing ne se fait pas « de configuration type en configuration type », mais bien au fil des

événements : une connexion sera ajoutée à tel événement puis, 5 événements après, c'est une autre qui sera ajoutée, et ainsi de suite.

Le routing se fait donc avec une *écriture persistante*. Conséquemment, et c'est une très grande difficulté qui se pose à l'analyste, il n'est pas possible de connaître la configuration du routing en lisant les écritures d'un événement – mis à part pour les tout premiers événements. Pour avoir cette configuration entière il est nécessaire de *remonter* les fichiers qlist, afin de savoir les derniers états des (29 au maximum !) connexions.

Pour effectuer notre travail d'analyse nous avons dû trouver une solution à ce problème. En effet, une des premières questions à laquelle nous devons répondre était « quels sont les modules qui produisent le son perçu ? »

Pour répondre à cette dernière nous avons développé un programme *ad hoc*¹²¹. Ce programme reçoit en entrée, successivement, tous les fichiers qlist et, après l'application d'un certain nombre de règles, il produit en sortie un fichier texte (1700 lignes), qui contient des représentations exhaustives du routing.

```
flute>harm>reverb>spat
flute>harm>spat
osc>reverb>spat
osc>spat
dac : harm,osc,reverb
```

Les lignes ci-dessus présentent la représentation du routing que nous avons conçu, à partir de *chaînes de routing*. Le programme réalisé est expliqué en profondeur au Chapitre VIII.

2.4.4 Représentation routing externe

La représentation du routing externe est donnée à la figure suivante. Les Modules de synthèse sont présents en orange, ceux de traitement en bleu clair.

¹²¹ La partition du compositeur ne pouvait suffire puisque les chaînes indiquées sont souvent uniques et ne réunissent que rarement plus de deux Modules.

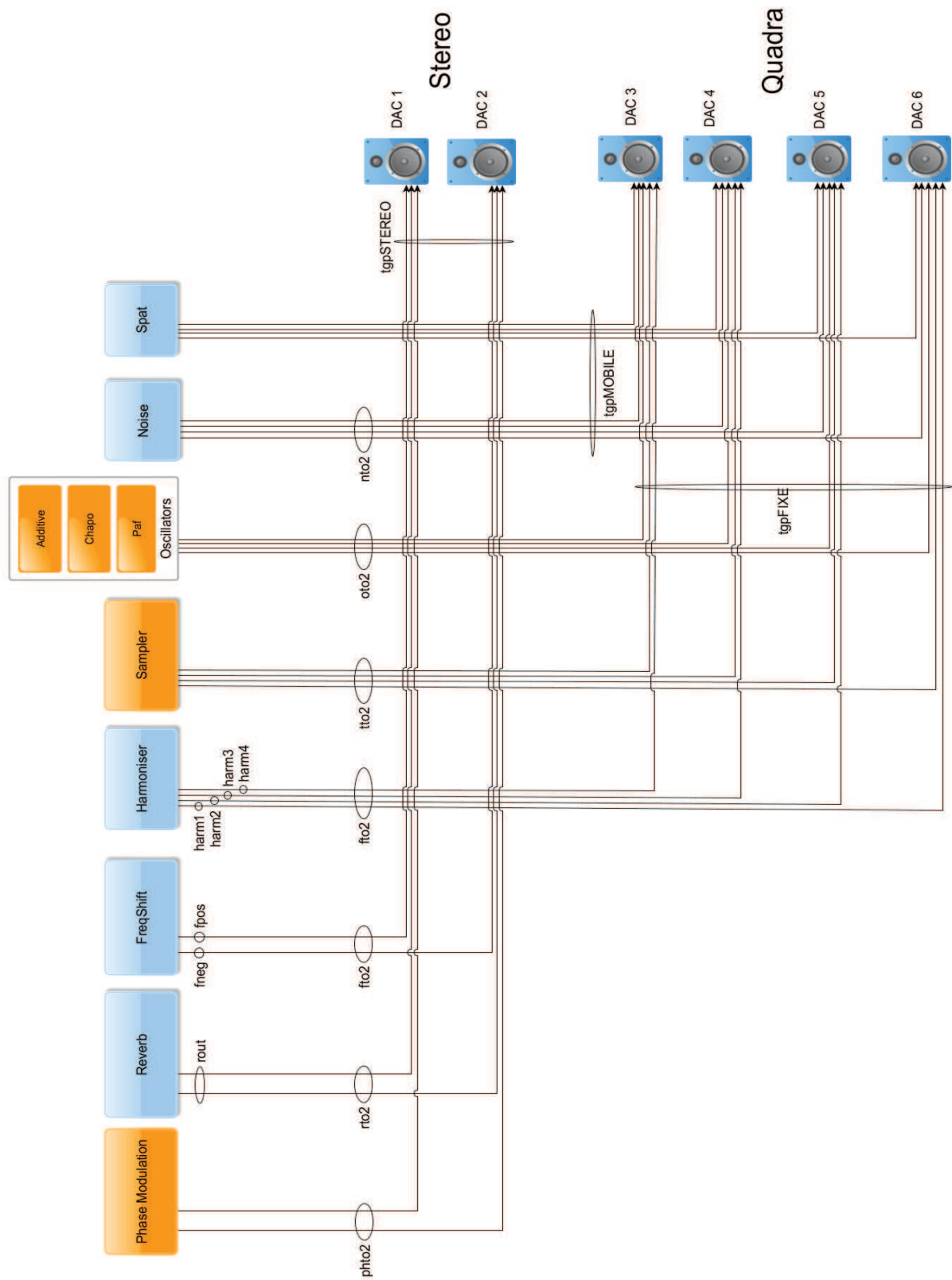


Fig. 20 - Le routing externe

Relevons que cette représentation est proche de l'implémentation réalisée, de la lecture du patch : le module de traitement Phase Shifting est intégré et le regroupement des trois modules de synthèse (*oscillators*) est présent. Aussi, dans une volonté de rendre compte des variables présentes dans l'artefact, nous avons inclus les paramètres qui contrôlent des niveaux d'amplitudes. Ils sont ici de trois types :

- Ceux qui contrôlent le niveau général en amplitude du module. Ils commencent par l'abréviation du module (e.g. *rout*, *fpos* ou encore *harm4*) ;
- Ceux qui contrôlent le niveau de sortie dans la diffusion. Ils commencent par l'abréviation du module et finissent par *to2* ;
- Ceux qui sont utilisés en direct lors du concert. Ils commencent par *tgp*.

3 Structuration de la pièce

Jupiter est une pièce conséquente d'une durée d'environ 30 minutes. Cette durée est une approximation car par le principe même de la musique en temps réel elle peut varier d'une interprétation à l'autre. Par exemple l'enregistrement sur lequel nous travaillons dure 28 minutes 30, et l'enregistrement de la flûte solo de Sophie Cherrier dure 27 minutes 47 secondes.

3.1.1 Sectionnement

Jupiter se divise en treize sections, nettement différenciées tant du côté de la partition que de celui de l'artefact. Pour la partition les sections sont marquées d'un chiffre romain, et pour l'artefact, comme on l'a vu, à chaque section correspond un fichier texte et pour la copie numérique et pour la partition électronique (les qlist). Le passage d'une section à l'autre se fait dans le patch à l'aide de la variable *section-number*¹²²

Comme nous le verrons ci-après ces sections sont loin d'être égales en durée. Certaines sont d'une durée de l'ordre de la minute et d'autres peuvent en valoir le triple.

3.1.2 Durée et événements

Le tableau suivant met en rapport la durée et le nombre d'événements des sections.

Section	Durée (min.sec)	Nb d'evt
1	3.15	109
2	1.47	82
3	2.43	151
4	2.02	26
5	3.24	27
6	3.26	34
7	0.45	18
8	2.18	28
9	1.53	16
10	0.47	16
11	1.08	20
12	3.03	33
13	1.53	27

Tab. 5 - Durée et nombre d'événements par section

¹²² Elle est appelée au dernier événement d'une section et prend simplement le numéro de la prochaine section.

Comme nous pouvons le lire il n'y a pas de correspondance entre durée et événements. Ceci apparaît plus clairement si nous représentons ces deux grandeurs avec un histogramme, comme le présente la figure ci-dessous.

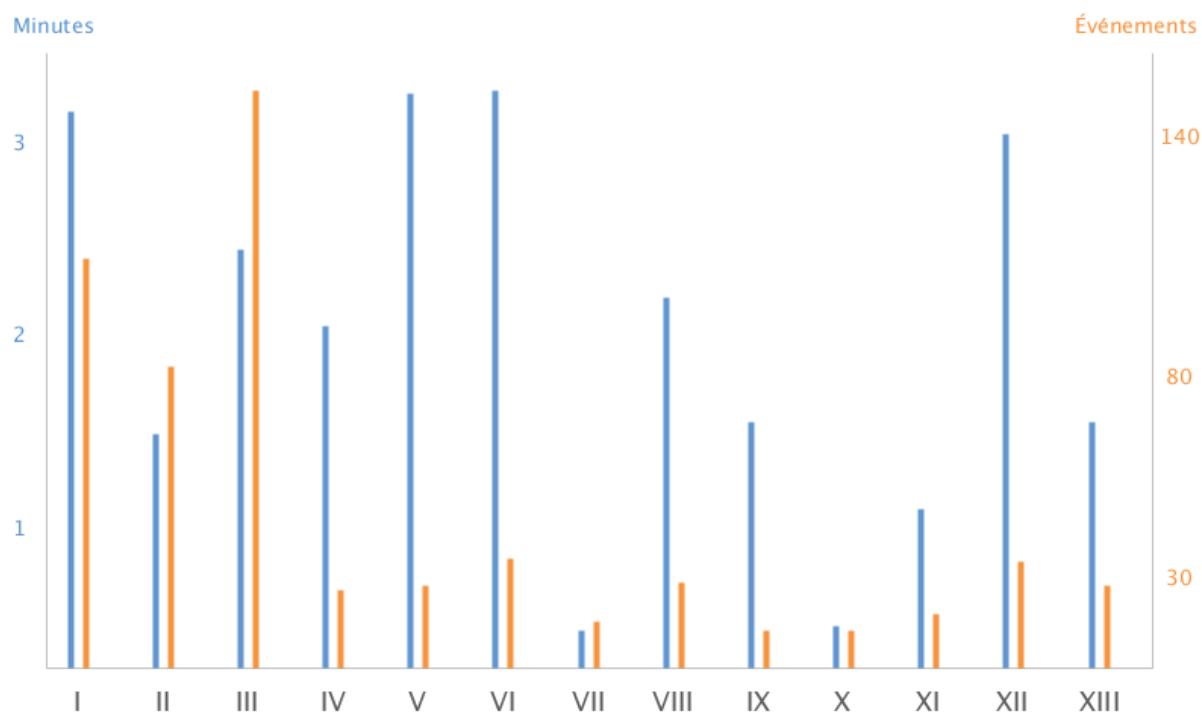


Fig. 21 - Comparaison de la durée et du nombre d'événements par section

Par exemple la section la plus longue, la VI, présente 34 événements, quand la section II, qui dure environ moitié moins, en présente 82.

Nous nous apercevons que les événements n'appartiennent pas à un niveau structurel précis mais que leur rôle structurel dépend de la composition. Les événements sont placés librement par le compositeur selon ses souhaits de synchronisation¹²³. Après notre travail d'analyse, nous voyons qu'ils se situent en majorité entre le niveau de la note (comme c'est le cas de la section III) et celui de la phrase (comme dans la section VI).

¹²³ Nous avons pu trouver des événements sans écriture.

Chapitre VII

Modules et Structure de contrôle de l'artefact

Résumé

Dans ce chapitre nous apportons la première réponse à notre problématique, nous mettons au clair la structure de contrôle de l'artefact. Ce chapitre est donc dédié aux 9 Modules. Il s'agit de les décrire afin de saisir leurs contrôles. La mise au clair de la structure de contrôle doit nous permettre par la suite de pouvoir lire et comprendre l'intégralité de la partition électronique. De la sorte, nous devrions être à même de comprendre comment la musique est faite, soit d'expliquer ce que l'on entend.

Essentiellement, ce chapitre concerne la description des Modules. Nous décrivons ceux de synthèse puis ceux de traitement, en prenant soin d'explicitier pour chacun la signification de leurs contrôles. Enfin nous apportons une vue globale sur la structure de contrôle de l'artefact, qui se dessine avec l'ensemble des contrôles.

1 Modules de synthèse

Dans ce grand titre les quatre Modules de synthèse – Sampler, Additive, Chapo et Paf – sont décrits en trois temps : (i) description de la méthode de synthèse, description le cas échéant de l'algorithme et description de l'implémentation réalisée, (ii) vue tabulaire sur tous les contrôles du Module et (iii) explication de(s) l'usage(s) effectué dans la composition.

Le premier temps peut varier conséquemment selon la présence ou non d'un algorithme et selon l'originalité de la méthode de synthèse – le Module Paf est ainsi plus détaillé que Additive. Le deuxième temps est lui identique à tous les Modules, il s'agit simplement de donner la signification des contrôles. Cette explication est en quelque sorte l'aboutissement du travail de description des Modules ; l'enjeu est de rendre compte de l'emprise sur le son et la musique que le compositeur s'est façonnée. Avec ces significations, avec leurs intégrations, il doit nous être possible de comprendre l'écriture des qlist.

1.1 Sampler

Le Sampler est un Module conséquent qui réunit deux utilisations nettement différentes. Il est constitué d'une base commune, le moteur audio, fait d'un ensemble de samples, réparti sur huit voies, avec une implémentation qui appelle 9 paramètres. À partir de cette base commune se distinguent deux utilisations. Une première, standard, où le Module est utilisé dans le cours du jeu de la flûte avec le contrôle *addsamp* ; une seconde, unique, où le Sampler rend sonore des séquences avec le contrôle *samp-note*, elles-mêmes engendrées par un algorithme d'interpolation. Comme cet algorithme n'est pas propre au Module mais qu'il concerne une utilisation, nous le détaillons dans le troisième temps dédié aux utilisations.

1.1.1 Implémentation audio

La méthode d'échantillonnage est utilisée avec une polyphonie de huit voies. Quatre familles de samples, tous instrumentaux, sont disponibles : Flûte, flûte accentuée avec la langue (Flute Tongue Ram), Piano, et Tam Tam. Afin de gagner en qualité de transposition chacune de ces familles contient plusieurs samples de différentes hauteurs¹²⁴. Alors, la difficulté de l'implémentation audio tient dans la sélection du sample approprié et, le cas échéant, de la transposition à effectuer pour obtenir la hauteur désirée. Ceci est effectué par un ensemble de calculs propres au designer que nous ne détaillons pas. Relevons simplement que l'implémentation fait appel à six paramètres : *pitch*, *velocity*, *duration*, *sample number*, *onset*¹²⁵, *risetime*, *decaytime* et *steadystate*. Les trois derniers plus *duration* permettent de paramétrer

¹²⁴ Flute Tongue Ram, Piano et Tam Tam contiennent chacun 6 samples et la famille Flûte en contient 13.

¹²⁵ Point de départ pour la lecture du sample.

une enveloppe d'amplitude, appliquée à chacune des voies. Sa forme et son paramétrage sont représentés dans la figure ci-dessous.

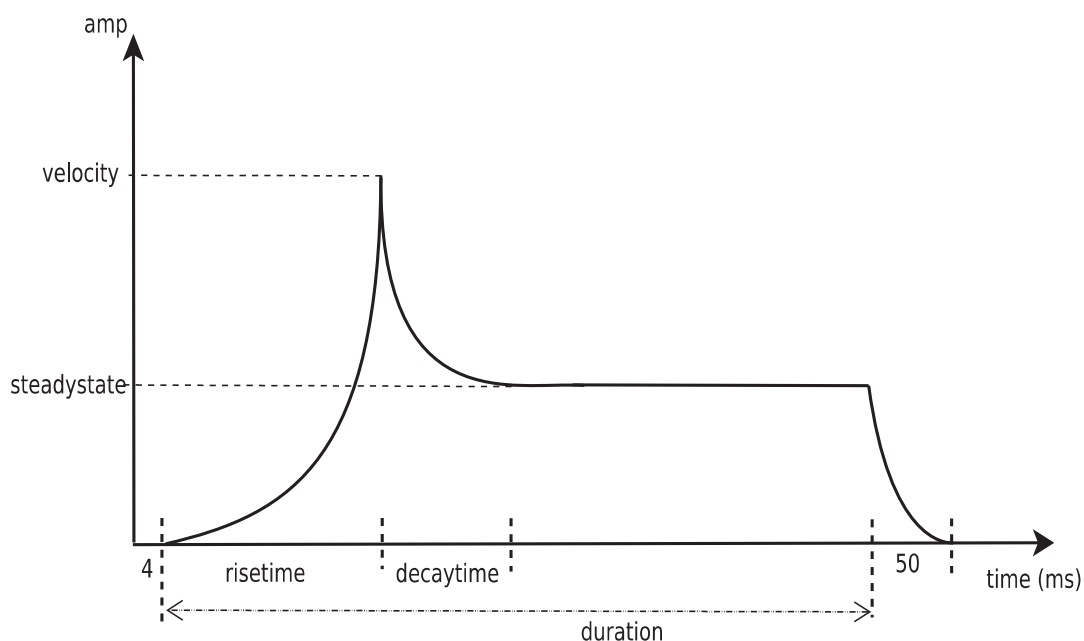


Fig. 22 - Paramètres de l'enveloppe d'amplitude appliquée sur chacune des voies du Sampler

1.1.2 Contrôles

C'est à partir de cette base commune que les deux utilisations vont se faire. Soit le Module est utilisé pour rendre sonore des séquences avec *samp-note*, soit il est utilisé au cours du jeu de la flûte avec les qlist via *addsamp*. Ces deux contrôles concatènent certains des précédents paramètres, comme le présente le tableau suivant.

Nom	Signification
<i>samp-note</i>	Variable de type liste avec la syntaxe suivante : pitch, velocity, duration
<i>addsamp</i>	Variable de type liste avec la syntaxe suivante : pitch, velocity, duration, onset, risetime, decaytime, steadystate

Tab.6 - Contrôles du Sampler

Relevons qu'aucun de ces contrôles n'inclut la sélection de samples. En effet, ceci est réalisé « en dur » dans le patch : pour *samp-note*, selon les séquences, ce sont les samples de Flûte Tomgue Ram, Piano ou de Tam Tam qui sont lus ; pour *addsamp* ce sont uniquement ceux de Flûte.

1.1.3 Utilisation avec *samp-note*

1.1.3.a Introduction

Le Sampler est ici utilisé pour rendre sonores des séquences. La lecture du patch et de quelques écrits (e.g. [May, 2006]) nous invite à différencier deux sortes de séquences selon qu'elles sont produites ou non par un algorithme d'interpolation. Certaines séquences, nommées *interpolations*, semblent en effet être générées par un algorithme dans le temps du concert (temps réel), quand d'autres, nommées *ostinato*, sont lues depuis des fichiers réalisés dans le temps de la composition (temps différé).

L'algorithme qui produit les séquences *interpolations* est le plus conséquent de l'artefact. Il doit permettre (i) d'enregistrer deux motifs rythmiques joués en direct par la flûte et (ii) d'effectuer un calcul d'interpolation, en un certain nombre de cycles, de sorte à obtenir de nouvelles séquences. Voici la vue en boîte noire d'un tel algorithme.

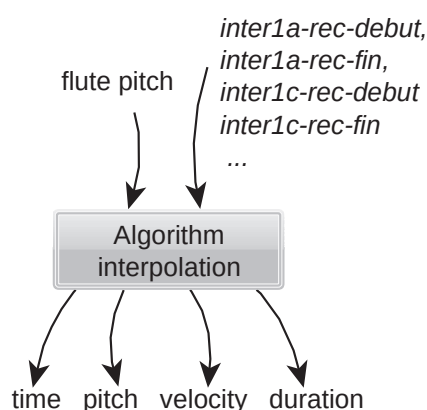


Fig. 23 - Vue en boîte de l'algorithme effectuant les *interpolations*

En entrée l'algorithme reçoit le flux de notes MIDI jouées à la flûte (*flute pitch*) et un ensemble de variables présentes dans les qlist, fonctionnant en binôme (e.g. *inter1a-rec-debut* et *inter1a-rec-fin*), qui déclenchent l'enregistrement des motifs rythmiques¹²⁶. En sortie l'algorithme produit des lignes de données avec la syntaxe {*time, pitch, velocity, duration*} qui constituent les séquences *interpolations*.

¹²⁶ Les premières variables de ces paires (e.g. *inter1a-rec-debut*) précisent de plus le nombre de notes à enregistrer et le nombre de cycles d'interpolation à effectuer – le nombre de notes étant identique pour les deux motifs.

Les idées véhiculées par un tel algorithme correspondraient ainsi à une musique du temps réel, où l'électronique varie avec les interprétations instrumentales – ici en effectuant des calculs à partir des durées de notes jouées¹²⁷.

Cependant, le lecteur l'aura compris, notre positionnement est ici différent. Nous faisons le choix de concevoir toutes les séquences comme étant réalisées dans le temps de la composition. Notre choix s'explique simplement par le fait que ce que nous pouvons lire de l'algorithme dans le patch est lacunaire ; d'après l'implémentation que nous avons lue (cf. Annexe A) il n'est en effet pas possible qu'un tel calcul puisse être réalisé dans le temps du concert. Par exemple, on trouve dans le patch des calculs pour effectuer des interpolations sur le plan rythmique, mais on ne trouve rien sur le plan des hauteurs. Or à l'écoute, ces séquences sont d'amples mouvements de l'aigu vers le grave ou inversement, très saillants. Un tel algorithme, s'il devait fonctionner « en live » devrait ainsi intégrer la combinaison des valeurs de durées avec celles de hauteurs, ce qui est absent du patch¹²⁸. Par contre, ce que l'on entend s'explique par quelques fichiers présents dans l'archive et chargés dès l'ouverture du patch, contenant la description exhaustive des séquences avec la même syntaxe { *time*, *pitch*, *velocity*, *duration* }.

L'algorithme qui engendre ces séquences interpolées avec d'amples mouvements d'ambitus existe bien, mais nous n'avons pas son implémentation entière.

Nous décrivons ci-après les deux types de séquences *interpolations* et *ostinato*. L'algorithme d'interpolation est décrit à l'aide des documents suivants : [Manoury, 2017]¹²⁹, [Manoury et al., 2003, Flûte-4x], [Manoury et al., 2003, Glossaire] et l'implémentation du patch qui, bien que parcellaire, reste éclairante.

1.1.3.b Aperçu global de l'algorithme

L'algorithme se subdivise en deux : un premier pour établir la répartition dans le temps des séquences et un second pour établir leur mouvement sur le plan des hauteurs. Comme nous le verrons, dans un cas comme dans l'autre, des calculs d'interpolation¹³⁰ sont effectués. Nous pouvons comprendre ces algorithmes dans une volonté de transgresser la théorie musicale occidentale. Sur le plan rythmique, en interpolant deux motifs, il résulte des organisations

127 Des notes de la flûte l'algorithme enregistre uniquement les durées, à l'aide de l'objet [timer], pour chaque changement de hauteur.

128 Trois points supplémentaires justifient notre choix : (i) les séquences sont polyphoniques or rien dans l'implémentation ne laisse apparaître une telle possibilité (une combinaison des données *time* et *duration* serait nécessaire) ; (ii) l'abstraction qui permettrait d'enregistrer les données des séquences produites ([explode]) ne reçoit pas d'instruction *record*. ; (iii) la valeur calculée à l'issue de l'interpolation des durées est utilisée une seule fois comme un temps de délai.

129 Accompagné d'un schéma réalisé par le compositeur : voir Annexe B.

130 Pour une compréhension de ce calcul nous renvoyons à [Lorrain, 2015] où un exemple est donné pour une interpolation rythmique. De façon abstraite, disons simplement ici que l'interpolation est un processus itératif qui permet de transformer un élément *a* en un élément *b* avec un certain nombre d'étapes (ou cycles).

temporelles qui ne peuvent être notées par la théorie¹³¹ ; sur le plan des hauteurs une nouvelle *carte de fréquences* est tracée qui va se substituer à celle de la gamme chromatique.

1.1.3.c Une carte de fréquences

Au sein de cette carte de fréquences l'algorithme permet de tracer ce que le compositeur appelle des *neumes*.

La réalisation de la carte part du fait que la gamme chromatique a pour étalon $2^{1/12}$. Alors, des étalons abstraits sont imaginés avec non plus 2 pour période mais b/a , un intervalle de fréquences, et n pour le nombre de tempéraments.

$$\text{étalon} = \left(\frac{b}{a}\right)^{1/n}$$

Afin de comprendre la conception de la carte imaginons ensuite deux étalons calculés selon les souhaits du compositeur, $\left(\frac{b}{a}\right)^{1/7}$ et $\left(\frac{d}{c}\right)^{1/7}$. La dernière étape consiste à trouver un moyen pour lier ces deux dimensions, de sorte à pouvoir passer successivement de l'une à l'autre et ainsi de construire une véritable carte. Ceci est réalisé par un calcul d'interpolation : les 7 divisions de l'intervalle $\frac{b}{a}$ et les 7 divisions de l'intervalle $\frac{d}{c}$ sont interpolées en 4 étapes, comme le présente le schéma suivant.

131 « Il est tout à fait impossible de noter en écriture traditionnelle les rythmes produits dans cette interpolation [rythmique] car leur complexité deviendrait insoluble. » [Manoury *et al.*, 2003, Flûte-4x, p. 12].

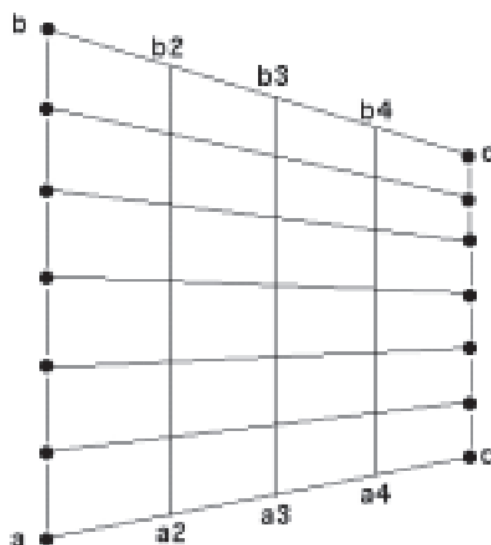


Fig. 24 - Carte de fréquences.
Capturé de [Manoury et al., 2003,
Glossaire]

Sur les axes horizontaux on trouve deux nouveaux étalons : sur l'axe [b, d] l'étalon $\left(\frac{b}{d}\right)^{1/4}$ et sur l'axe [a, c] l'étalon $\left(\frac{c}{a}\right)^{1/4}$. Sur les axes verticaux on trouve 4 nouveaux intervalles, [b2, a2], [b3, a3] et [b4, a4], chacun divisé en 7 tempéraments.

L'idée des *neumes*, par métaphore, consiste alors simplement à tracer des chemins ascendants et descendants avec des points adjacents de la carte.

1.1.3.d Des interpolations rythmiques

Les interpolations rythmiques sont effectuées à partir de deux motifs, A et B, écrits pour la flûte. Le calcul consiste alors à transformer successivement le motif A en B en un certain nombre d'étapes¹³². Par exemple, la première interpolation présente dans la section V interpole deux motifs en 14 étapes, comme le présente la partition traditionnelle.

¹³² Le calcul d'interpolation du patch Pure Data est le suivant : $Out = dur1 * (1-inc) + dur2 * inc + 0,5$
avec *out* la durée de la nouvelle note ;
dur1 la durée de la note du motif A ;
inc, compris entre 0 et 1, le coefficient du numéro de l'étape en cours sur le nombre d'étape à réaliser ;
dur2 la durée de la note du motif B.



Fig. 25 - Représentation de la première interpolation (inter1a) dans la partition gravée

En haut est représenté ce qui rentre dans l'algorithme, c'est-à-dire les deux motifs rythmiques ; en bas ce qui en sort, à savoir la séquence interpolée sur le plan rythmique faite d'un tas de chemins dans une nouvelle carte de fréquences.

1.1.3.e Conclusion, fichiers d'interpolation et d'ostinato

La pièce intègre au total 16 séquences dont 7 *interpolations* et 9 *ostinatos*. Toutes sont présentes sous forme de fichiers placés dans l'archive. Dans les lignes de données ci-dessous se trouvent, à gauche, un extrait d'un fichier d'interpolation (*interia*), et à droite un extrait d'un fichier d'ostinato (*ostri*).

1330	6039	127	1954	100	4374	110	3200
330	6078	127	1910	100	4839	110	3200
1080	5999	127	2000	100	5303	110	3200
1600	6276	127	1705	100	5767	110	3200
660	5000	0	22446	100	6232	110	3200

C'est donc la même syntaxe qui est utilisée. La première colonne correspond au laps de temps (ms) avant la lecture de la ligne suivante, et les trois autres colonnes correspondent à la syntaxe de *samp-note*, à savoir *pitch* (en MIDI cents), *velocity* et *duration* (ms).

Deux différences peuvent être relevées entre ces fichiers. Pour les *ostinatos* le temps entre deux impulsions est soit égal à 100 soit dégressif d'une ligne à l'autre de sorte à former un *accelerando*, tandis que pour les *interpolations* il n'y a pas de règle apparente. La seconde différence, plus simple, concerne la vitesse : pour les *interpolations* la vitesse est toujours maximale (ou nulle), tandis que pour les *ostinato* toute la profondeur est utilisée [0, 127].

La lecture des fichiers se fait d'abord en chargeant leur contenu dans des structures de données de Pure Data (cf. Annexe A pour les captures d'écran du patch), et ensuite avec des variables *ad hoc* présentes dans les qlist qui les déclenchent. Par exemple *interia-start* lance la

lecture de la première interpolation et *ostri-start* la première séquence *ostinato*. Le tableau ci-dessous décrit ces 16 séquences.

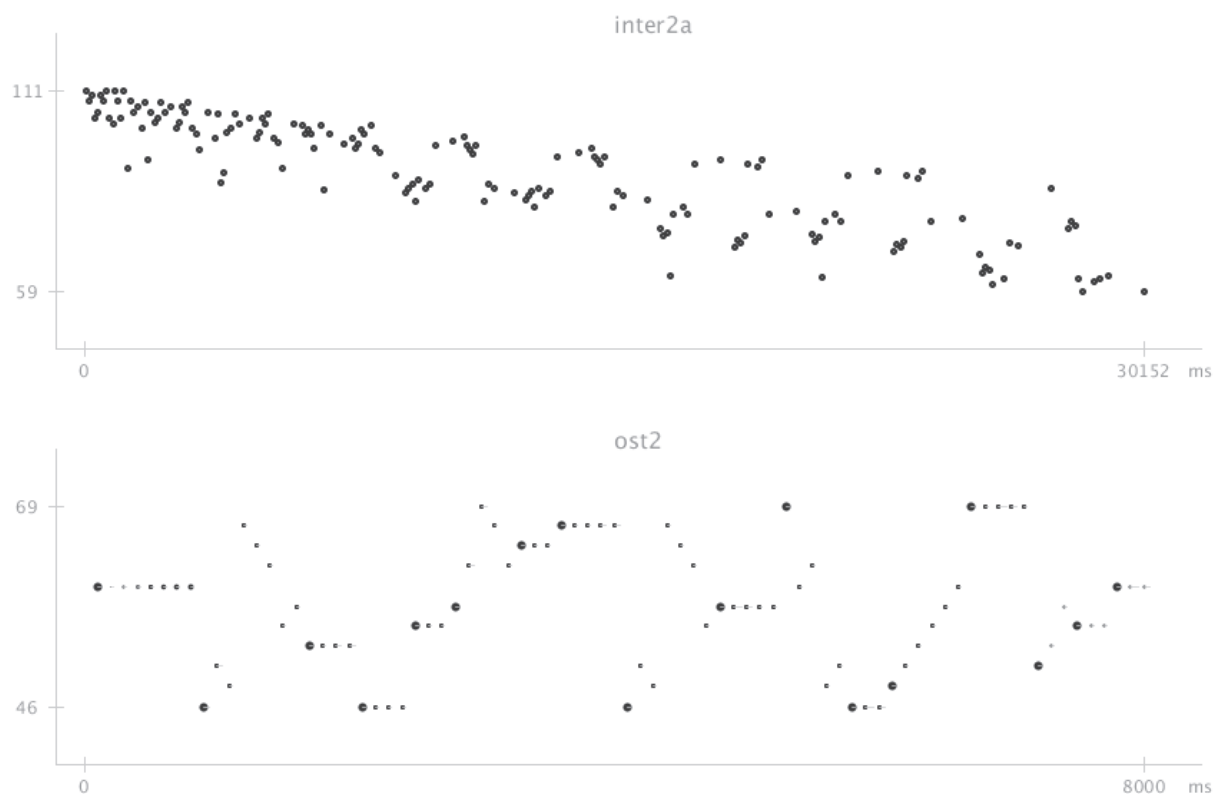
Nom	Section	Sample	Nb de cycles	Nb d'impulsions	Durée (sec)
Inter1a	V	Piano	14	96	~ 50
inter1c	V	Piano	14	208	~ 55
inter1d	V	Piano	4	150	~ 54
inter2a	IX	Piano	16	162	~ 30
inter2b	IX	Piano	16	162	~ 30
inter2c	IX	Piano	? ¹³³	486	~ 28
inter2d	IX	Piano	2	40	~ 12
ost1	VIII	Tam Tam	∅	44	~ 4
ost2	VIII	Tam Tam	∅	80	~ 8
ost3	VIII	Tam Tam	∅	71	~ 7
ost4	VIII	Tam Tam	∅	551	~ 5
ost4b	VIII	Tam Tam	∅	213	~ 28
ost5	X	FI Tongue	∅	176	~ 17
ost6	X	FI Tongue	∅	16	~ 1.5
ost7	X	FI Tongue	∅	44	~ 3
ost8	X	FI Tongue	∅	300	~ 17

Tab. 7 - Description de toutes les séquences

1.1.3.f Représentation des séquences

Comme nous possédons toutes les données qui décrivent ces séquences nous pouvons les représenter dans un espace usuel, à deux dimensions, temps-fréquence. Les lignes des fichiers sont alors à concevoir comme des impulsions, représentées par des points situés dans l'espace temps-fréquence. La vélocité est elle représentée par la taille des points et la durée, le cas échéant, par une fine ligne.

¹³³ La séquence *inter2c* ne reçoit pas d'instruction d'enregistrement dans les qlist ; elle est donc conçue dans le temps de la composition (qlist vérifié dans Pure Data et dans Max).



*Fig. 26 - Représentations des séquences.
En haut une séquence d'interpolation ; en bas une ostinato*

Ces représentations, qui aident l'écoute, nous ont été fortement utiles pour l'analyse ; elles sont ainsi exposées au cours du chapitre IX.

1.1.4 Utilisation avec *addsamp*

Il s'agit d'une utilisation nettement plus courante où les écritures sont présentes dans les qlist. Deux sous-utilisations sont à différencier. Une première où le Sampler fonctionne de pair avec un autre Module, Additive, et une seconde où des notes polaires sont installées. Afin de ne pas surcharger nous décrivons ici les notes polaires et laissons l'utilisation avec Additive au prochain Module.

Les notes polaires ont ceci de particulier qu'elles sont prescrites par l'intermédiaire du patch. Une vingtaine de variables à usage unique sont ainsi présentes dans les qlist (e.g. *sec2-evt77*) qui activent des métronomes et/ou des *delay*, qui eux-mêmes activent *addsamp*, comme le présente la figure ci-dessous.

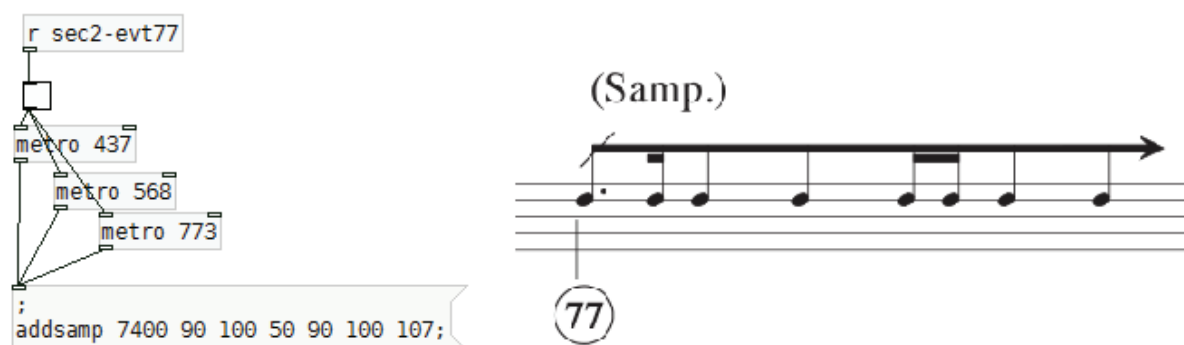


Fig. 27 - Note polaire déclenchée par l'intermédiaire du patch.
A gauche le patch Pure Data ; à droite la notation musicale extraite de la partition.
section II, événement 77

Ces notes polaires viennent en général appuyer une note de la flûte, la prolonger. Elles sont présentes uniquement au début de la pièce, localement dans la section III et dans la coda de la II.

1.2 Additive

Ce Module utilise la méthode additive avec 20 oscillateurs, répartis équitablement sur 4 voies. C'est un Module utilisé très localement au début de la pièce dans une partie de la section III¹³⁴ et en parallèle du Sampler. Il se contrôle avec une variable de type liste, *addsyn*.

1.2.1 Description

Les 20 oscillateurs peuvent lire 4 formes d'onde différentes¹³⁵ accessibles par un numéro, comme le présente la table ci-dessous.

Numéro	Nb de partiels	Distribution des partiels
4	4	p1, 3*p1, 7*p1, 8*p1
6	2	p1, 6*p1
7	3	p1, 2*p1, 3*p1

Tab. 8 - Variables de contrôle de Additive, concaténée dans *addsyn*

En supplément un vibrato et une enveloppe d'amplitude sont ajoutés sur chacun des oscillateurs. La figure ci-dessous présente le schéma de synthèse pour une voie.

¹³⁴ Dans le patch il est ainsi nommé « sect-3 »

¹³⁵ Dans l'implémentation on voit que 8 tables sont nécessitées. Seulement, après analyse des qlist (cf. Chap VIII 1.2), on se rend compte que seules quatre sont utilisées.

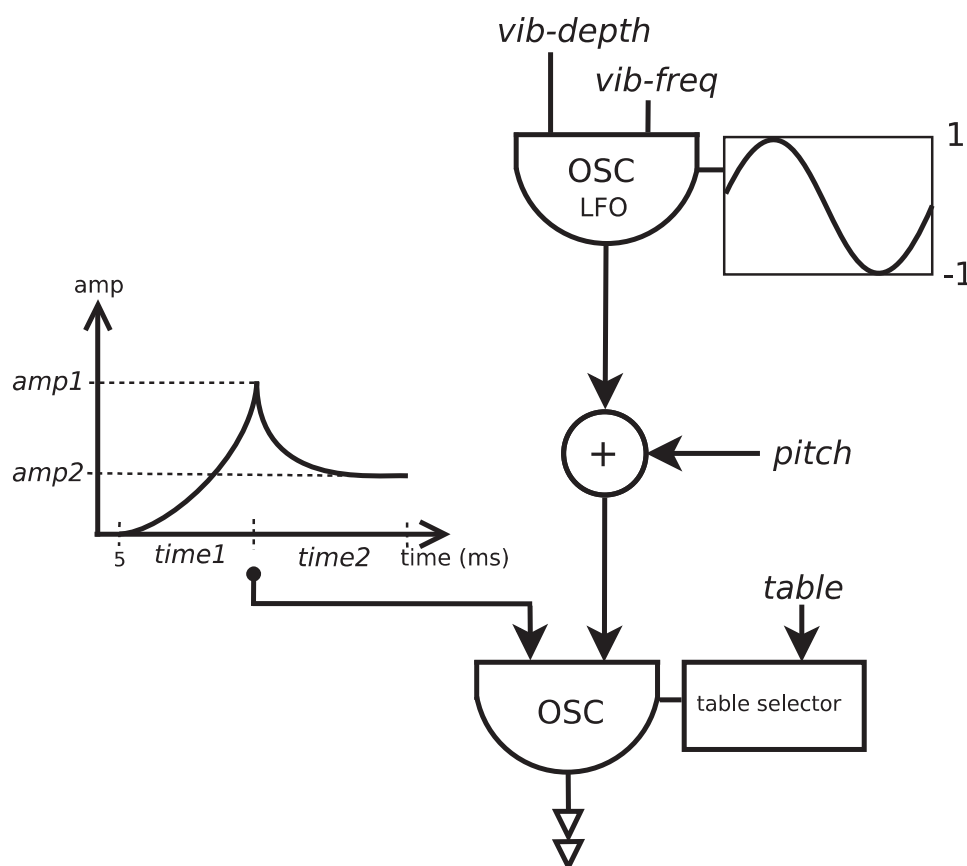


Fig. 28 - Schéma de synthèse du Module Additive. Les variables présentes dans le qlist sont indiquées en italique

1.2.2 Contrôle

Le contrôle se fait à l'aide d'une seule variable, *addyn*, qui concatène 8 autres variables avec la syntaxe suivante : *channel*, *amp1*, *time1*, *amp2*, *time2*, *pitch*, *vib-depth*, *vib-freq*, *table*. *channel* correspond au numéro de l'oscillateur ; *amp1*, *time1*, *amp2*, *time2* correspond au paramétrage de l'enveloppe d'amplitude (cf. figure précédente) ; *pitch* indique la fréquence de l'oscillateur en MIDI ; *vib-depth* et *vib-freq* correspondent à la profondeur et la fréquence du vibrato et *table* indique le numéro de table.

1.2.3 Utilisation

Comme nous l'avons vu Additive est utilisé de pair avec le Sampler dans une partie de la section III (evt 65 - 100). Dans ce passage, les Modules engendrent des gestes électroniques scintillants, très fluides, situés dans le medium, que nous nommons dans l'analyse des *vaguelettes*. Elles se constituent d'une dizaine de « notes » chacune produite par le Sampler et par Additive, comme l'illustre l'extrait des qlist ci-dessous.


```

addsamp 7500 156 1000 90 90 150 110;
addsyn 1 127 10 50 1000 7500 1.5 12. 4;
100 addsamp 7300 156 1000 90 90 150 110;
addsyn 2 127 10 50 1000 7300 1.5 12. 4;
100 addsamp 6900 156 1000 90 90 150 110;
addsyn 3 127 10 50 1000 6900 1.5 12. 4;
100 addsamp 6400 156 1000 90 90 150 110;
addsyn 4 127 10 50 1000 6400 1.5 12. 4;
100 addsamp 6200 156 1000 90 90 150 110;
addsyn 5 127 10 50 1000 6200 1.5 12. 4;
100 addsamp 5800 156 1000 90 90 150 110;
addsyn 6 127 10 50 1000 5800 1.5 12. 4;
100 addsamp 5400 156 1000 90 90 150 110;
addsyn 7 127 10 50 1000 5400 1.5 12. 4;
addsyn 9 0 100 0 100;
addsyn 10 0 100 0 100;
addsyn 11 0 100 0 100;
addsyn 12 0 100 0 100;
addsyn 13 0 100 0 100;
addsyn 14 0 100 0 100;
addsyn 15 0 100 0 100;

```

Les notes sont donc séparées de 100 ms et c'est bien la même hauteur qui est attribuée aux deux Modules. Il y a parallèlement un jeu dans la succession de ces notes entre les Modules. Additive est utilisé pour l'attaque (127 en 10 ms) tandis que le Sampler, avec une attaque de 90 ms, l'est davantage pour le maintien : la note suivante arrive donc sur le maintien de la précédente et ainsi de suite. Ce maintien contribue à l'effet de fluidité. Relevons aussi une différence forte entre le Sampler et Additive : tandis que les sons du premier ont nécessairement une fin, ceux du second sont maintenus *ad lib*. En lisant la suite du précédent extrait dans les qlist nous voyons ainsi que les 6 voies activées d'Additive sont maintenues¹³⁶ jusqu'à la 3^e vaguelette. Entre les vaguelettes il n'y a donc pas de silence mais le maintien des sinusoïdes d'Additive – que la Reverb prolonge de plus – ce qui contribue à cette fluidité, et à un rendu épuré.

1.3 Chapo

Ce module est particulier puisque d'une part il intègre le second algorithme de la pièce, et d'autre part c'est le seul Module de synthèse à utiliser directement l'échelle des fréquences pour investir le spectre – et non la norme MIDI.

L'idée musicale qui correspond à l'algorithme s'inscrit au cœur de la *musique en temps réel*. Il s'agit de tisser un lien entre les hauteurs de notes jouées par la flûte et le son électronique

¹³⁶ Dans l'extrait donné l'extinction des voies 9 à 15 s'avère en fait inutile car elles n'ont pas été activées auparavant.

produit par le Module ; soit d'établir une interaction entre instrumentiste et électronique. Mais cette interaction est différente de celle issue du suiveur de partition : quand cette dernière est bâtie sur des événements – donc une interaction fondamentalement discrétisée – l'interaction du Chapo se fait au cours du jeu de la flûte, *continuellement*.

L'écriture du Module en fréquence appelle inévitablement un changement dans la façon de composer. Pour prescrire ces valeurs en fréquence, pour investir le domaine fréquentiel, un procédé spécifique est ainsi employé. Nous le détaillons après avoir rendu compte de la synthèse et de l'algorithme.

1.3.1 Description globale

La réalisation de l'idée musicale consiste à apposer une *enveloppe spectrale* sur un stack de 28 sinusoïdes¹³⁷, et ensuite de déplacer cette enveloppe en fonction des hauteurs de notes jouées à la flûte. L'interaction se fait donc entre la position de l'enveloppe spectrale et la hauteur du son de flûte. Comme l'enveloppe se déplace continuellement dans le spectre, nous parlons par la suite de balayement.

1.3.1.a Algorithme et analyse de la flûte

Les notes MIDI de la flûte issues de l'analyse – via l'objet [fiddle~] – sont reçues à l'entrée de l'algorithme. En plus de cette variable, que nous nommons *flutePitch*, deux paramètres sont implémentés, *chat* et *porta*. C'est avec ces trois derniers que l'algorithme calcule une nouvelle variable que nous nommons le *facteur f*.

1.3.1.b Oscillateurs

Ils peuvent lire deux formes d'ondes, soit une sinusoïde, soit une faite de deux partiels – un fondamental et un harmonique de rang trois. La spécification de la table est globale, elle se fait avec la variable *chapo-tab*.

L'affectation en fréquence des oscillateurs est simple, elle se fait avec *chapo*, où en premier est indiqué le numéro de l'oscillateur suivi de sa fréquence, comme le présente l'extrait ci-dessous (section 6 evt 2) :

```
chapo 0 32.700001;
chapo 1 187.779999;
chapo 2 142.;
chapo 3 297.779999;
chapo 4 261.619995;
chapo 5 407.779999;
chapo 6 362.;
chapo 7 517.779968;
```

¹³⁷ À l'instar d'un cluster pour des notes, nous employons le terme stack pour des sinusoïdes

L'amplitude des oscillateurs ne se fait pas manuellement, elle n'est pas écrite, mais déduite de l'enveloppe spectrale. En fait, le déplacement de cette dernière, au cours du jeu de la flûte, revient à adresser dans le même temps une *enveloppe d'amplitude* à chacun des 28 oscillateurs.

Dès lors, et comme nous avons parlé d'interaction *continue*, il se pose la question de la résolution de ces enveloppes d'amplitude : quelle est leur fréquence de définition ? Avant de répondre à cette question il nous est nécessaire de décrire l'algorithme et l'implémentation audio.

1.3.1.c Enveloppe spectrale

Elle est simplement réalisée par une table de données, *chapo_spect*, qui associe pour un index compris entre 0 et 127 des valeurs comprises entre 0 et 126, avec la répartition suivante.

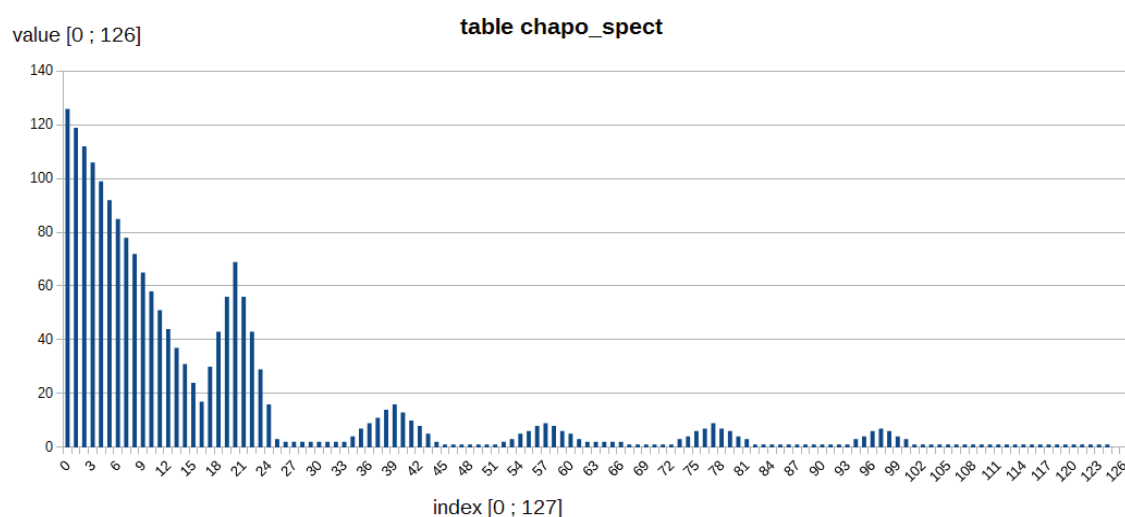


Fig. 29 - L'enveloppe spectrale de Chapo

L'index de la table est simplement calculé par une multiplication entre le *facteur f* et la fréquence adressée à l'oscillateur. En mettant de côté le *facteur f*, nous pouvons déjà relever deux points :

- le stack de sinusoides n'est jamais diffusé dans son entièreté ; une importante partie des sinusoides a donc une amplitude nulle
- la distribution des amplitudes est telle que l'on retrouve *grosso-modo* un spectre proche des instruments traditionnels : les premiers partiels ont une amplitude inversement proportionnelle à leurs fréquences, jusqu'à l'apparition de cinq formants, répartis sur le long de l'enveloppe.

La figure ci-dessous représente l'ensemble du Module, entre stack de sinusoides, enveloppe spectrale et algorithme.

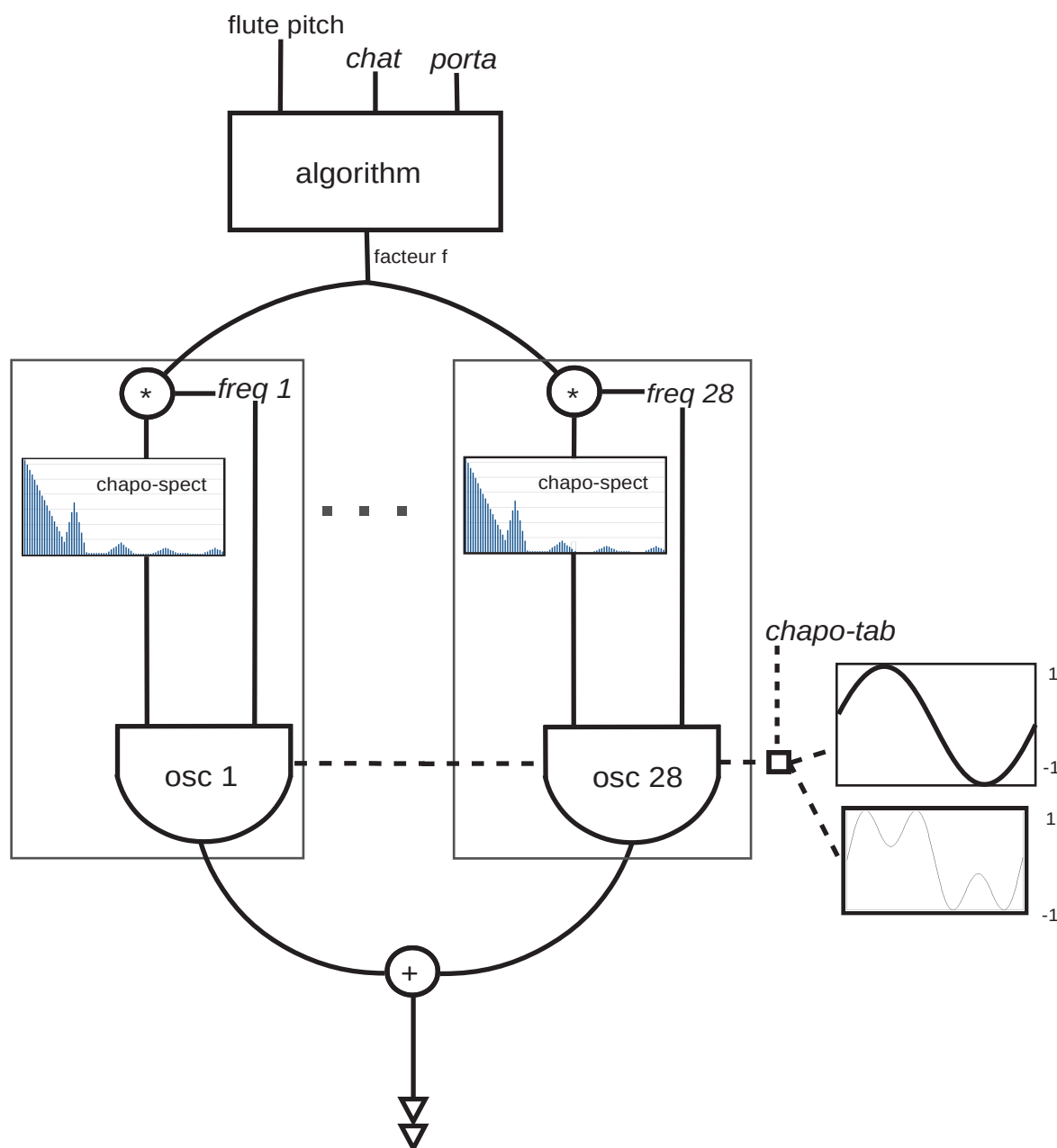


Fig. 30 - schéma de synthèse de chapo.
 Les variables en italique sont celles présentes dans les qlist

1.3.2 Algorithme

L'algorithme est fait de deux étapes. D'abord il s'agit de fixer le rapport entre les variations de hauteur de la flûte et l'amplitude générale du signal produit (la variable *chat*), et ensuite il s'agit de contrôler les évolutions temporelles de l'enveloppe spectrale (la variable *porta*).

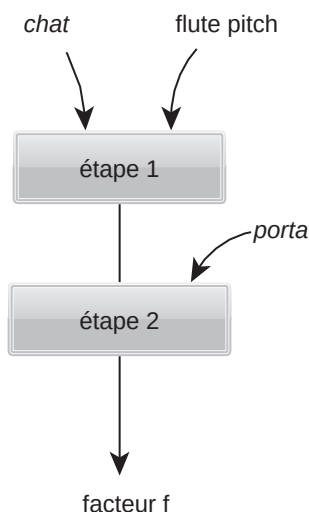


Fig. 31 - Les étapes de l'algorithme du Chapo.
Les variables en taliques sont celles présentes dans les qlsit.

Étape 1

Elle fixe le rapport entre les hauteurs de la flûte et le *facteur f*. Les calculs sont tels que le *facteur f* est inversement proportionnel à la hauteur de la flûte. Soit, plus la note est aiguë, plus le *facteur f* est faible, donc, en regard de la table, plus les amplitudes des oscillateurs sont conséquentes.

La variable *chat* permet de contrôler la variation du *facteur f*. Une valeur faible entraîne des variations élevées du *facteur f*, donc peu voire pas d'amplitude, et inversement. Ainsi la variable *chat* permet simplement d'avoir un contrôle sur l'amplitude globale du signal produit.

Étape 2

A l'issue de la première étape, le *facteur f* évolue en synchronicité avec les hauteurs de la flûte : à chaque changement de hauteur une nouvelle position de l'enveloppe spectrale est donnée. Cette seconde étape permet de nuancer cette synchronicité. Elle autorise un lissage du *facteur f* : un balayement de l'enveloppe spectrale. Les valeurs du *facteur f* vont ainsi devenir des points de destination à atteindre ; nous parlerons donc de balistique.

Le lissage est effectué par un simple système avec boucle de rétroaction, de sorte que la valeur sortante est calculée en fonction de la valeur entrante et de celle sortante :

$$Output = output + k(input - output)$$

où k est un facteur déterminé par la variable *porta*

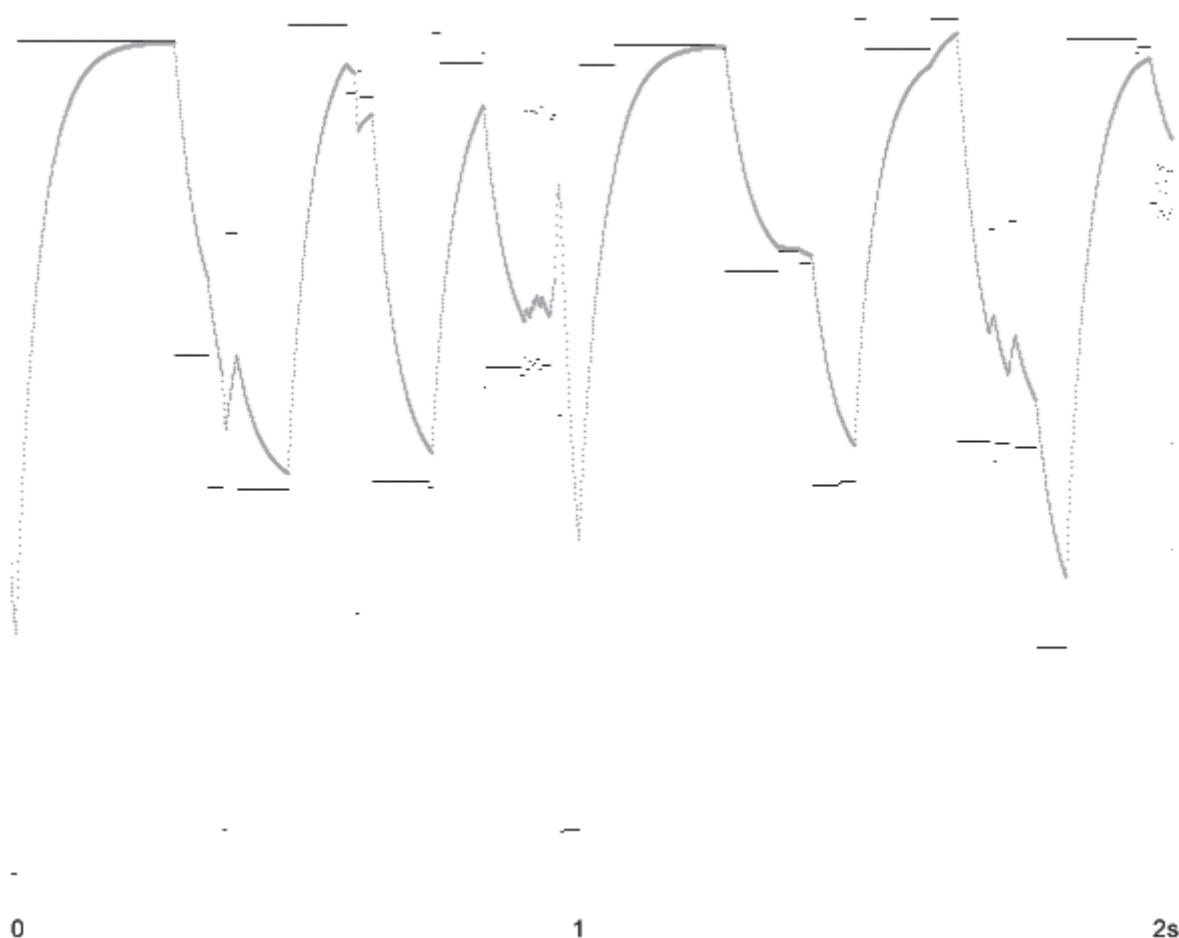
On peut maintenant répondre à la question de résolution des enveloppes d'amplitude. L'algorithme calcule le *facteur f* toutes les 20 ms¹³⁸. Ainsi, que la flûte joue ou non, les amplitudes des oscillateurs sont de nouveau calculées à cette fréquence.

Ce léger système balistique se contrôle avec la variable *porta*. Elle peut être vue comme la définition d'un *nombre de palier* à insérer entre la valeur de destination (in) et la valeur sortante (out). Par exemple, à l'instant t_0 , avec un nombre de paliers égal à 100 entre la valeur de destination et celle sortante, la nouvelle valeur sortante correspondra au premier palier de cette centaine. 20 ms après, au temps t_1 , la différence est recalculée et de nouveau 100 paliers sont affectés dans cette différence, et la valeur sortante sera de nouveau le premier palier de la centaine représentant cette différence qui s'est amoindrie, et ainsi de suite.

Nous devinons ici le comportement du système balistique : plus la valeur de destination est loin de la valeur sortante, plus le changement du *facteur f* est grand, et inversement.

En prélevant au cours de l'exécution du patch le *facteur f* et la valeur entrante dans cette seconde étape, il est possible de représenter ce comportement. Précisons qu'il ne s'agit pas de relever les *valeurs* des deux variables, mais bien de rendre compte du *comportement* balistique. La représentation ci-après ne possède donc pas d'unité sur ces axes.

138 C'est un [metro] qui actualise l'algorithme à cette fréquence.



*Fig. 32 - relevé du facteur f
 porta = 1024
 abscisse : temps (sec) ; ordonnée : facteur f
 noire : valeur de destination ; gris : facteur f*

Le système balistique simule donc une fonction logarithmique.

1.3.3 Interaction flûte et spectre

L'interaction entre la flûte et le signal est limpide : des notes aiguës de la flûte entraînent un déplacement de l'enveloppe vers le bas, donc un spectre résultant plus riche, et des notes graves entraînent un balayement vers le haut, donc un spectre résultant moins riche.

Ce qui importe le plus pour ce balayement, très perceptif, c'est l'importance des intervalles joués par la flûte. Plus un intervalle est conséquent plus le balayement est audible, et inversement.

Le contrôle *porta* s'avère maintenant essentiel : il fixe le temps de lissage de l'enveloppe spectrale. Au regard des sections VI et XII, il sera parfois nul – l'enveloppe sera alors synchrone

à la flûte – et parfois important (1024 paliers¹³⁹) – l'enveloppe se déplacera alors dans le spectre, ce qui se rapproche d'un effet de *portamento*.

1.3.4 Contrôle

En plus des quatre variables que nous avons vues, l'implémentation est telle qu'une prescription nulle en fréquence à un oscillateur engendre une amplitude nulle. De cette façon, le compositeur peut doser le nombre d'oscillateurs à activer.

Nom de variable	Signification
chapo	Variable de type liste : numéro d'oscillateur, fréquence (Hz)
chat	Amplitude générale du Module
porta	Temps de lissage de l'enveloppe spectrale
chapo-tab	Sélection de formes d'onde des oscillateurs

Tab. 9 - Contrôle du Chapo

1.3.5 Utilisation

Deux sections sont dédiées au Chapo, la VI et la XII. Dans celles-ci le Chapo est continuellement actif de sorte que l'interaction avec la flûte puisse se dérouler. Parallèlement ces deux sections se démarquent par leur structure, plus limpide, faite d'une quinzaine de phrases bien scindées. Dans chaque phrase le Chapo est utilisé à deux reprises : dès l'ouverture avec une nouvelle distribution de fréquences (la variable *chapo*), puis dans l'écoulement des phrases avec le temps de synchronisation de l'enveloppe spectrale (la variable *porta*), où l'on passe d'une synchronisation instantanée à une plus fluide avec des balayements qui se laissent entendre.

1.3.5.a Écriture verticale des fréquences

Comme nous l'avons introduit, Chapo est particulier puisque son écriture fait appel à des fréquences et non à des notes MIDI. En parallèle d'une écriture dans le temps de la pièce, il y a donc une écriture verticale qui superpose des sinusoïdes. Nous présentons ici la règle que nous avons trouvée qui explique ces écritures.

La distribution des fréquences se fait directement dans les qlist, via *chapo*, à chaque début de phrase, comme le présente l'extrait ci-dessous (section VI, evt 1).

```
chapo 1 155.559998;
chapo 2 77.779999;
chapo 3 233.080002;
```

¹³⁹ On remarque à la lecture des qlist que *porta* a toujours des valeurs en puissance de deux, comme s'il s'agissait d'une fenêtre temporelle.


```

chapo 4 155.559998;
chapo 5 311.130005;
chapo 6 233.080002;
chapo 7 388.779999;
chapo 8 311.130005;
chapo 9 466.160004;

```

La lecture des 17 stack présents dans la section VI laisse apparaître une certaine règle quant à leurs distributions. Celle-ci peut être comprise comme la volonté de donner au spectre produit des propriétés à la fois harmonique et inharmonique. En effet les partiels, à l'instar des spectres des instruments traditionnels, sont bien distribués par rang, mais ils s'en différencient puisqu'ils ne sont pas des multiples entiers d'une même fréquence.

La distribution des fréquences se fait avec deux séries de 14 fréquences. Pour chaque série une fréquence de référence (*fref*) et deux coefficients multiplicateurs (k_1 et $k_2 \in]0 ; 1[$) sont fixés. Les fréquences des séries sont alors calculées de la façon suivante, avec k_1 pour la première et k_2 pour la seconde.

```

p1 = k1 * fref
p2 = (k1+1) * fref
p3 = (k1+2) * fref
p4 = (k1+3) * fref
...
p14 = (k1+14) * fref

```

La proximité avec l'harmonicité vient du fait que pour chaque série la différence de fréquences entre les partiels est la même ; la proximité avec l'inharmonicité est double : d'abord *fref* n'est pas présent dans le spectre et ensuite les coefficients k_1 et k_2 sont des nombres non entiers.

Cette règle n'est cependant pas appliquée avec rigueur et l'on trouve au contraire des altérations quasi systématiques. Trois sortes d'altérations, qui tendent soit vers l'harmonicité soit vers l'inharmonicité, sont présentes : (i) souvent les deux premiers oscillateurs ont pour fréquences un des premiers harmoniques de la note jouée par la flûte ; (ii) il arrive souvent que les séries soient construites avec plusieurs coefficients, afin de venir brouiller cette répartition ; puis (iii) parmi la trentaine de stack qui forment ces sections il y a en 4 qui sont pleinement harmoniques¹⁴⁰.

¹⁴⁰ Le premier oscillateur reçoit alors un fondamental (p_1) et tous les autres sont des multiples entiers de ce dernier.

1.3.5.b Écriture horizontale

En supplément de cette précédente règle nous avons trouvé un schéma utilisé pour l'écriture dans le temps des sections. Un tel schéma devrait rendre compte de la structuration des sections. Néanmoins, après écoute, nous nous sommes aperçus qu'il s'agissait d'un travail poétique que l'on ne retrouve pas dans la perception, dans l'esthétique. Pour cette raison nous présentons ce schéma dans ce chapitre et non dans le suivant.

Le schéma utilisé permet de distribuer les *fref* des stack en fréquences des deux sections. Il s'agit simplement d'une suite de quatre notes ascendantes où *fref* prend la valeur du fondamental. Pour la section VI c'est la suite {Ré#₁, La₁, Ré₂, Sol₂}, répétée quatre fois¹⁴¹ ; pour la XII c'est la suite {La₂, La#₂, Ré#₃, Mi₃} répétée aussi quatre fois¹⁴².

Ce schéma n'empêche pas les stack de fréquences d'être disparates. En effet les coefficients changent eux à chaque stack – si ce n'est à l'intérieur d'une même série comme nous l'avons vu – et de plus les fréquences des premiers oscillateurs sont placées librement selon le jeu de la flûte.

La figure suivante représente les distributions effectuées verticalement et horizontalement pour la section VI.

¹⁴¹ *fref* prend donc successivement les valeurs {77.78, 110, 147, 196}, répétées quatre fois ; pour la dernière phrase cependant *fref* vaut Sol₂ (196).

¹⁴² *fref* prend donc successivement les valeurs {220, 233, 311, 330}, répétées quatre fois ; cependant pour la 16^e phrase *fref* vaut La₂ (200).

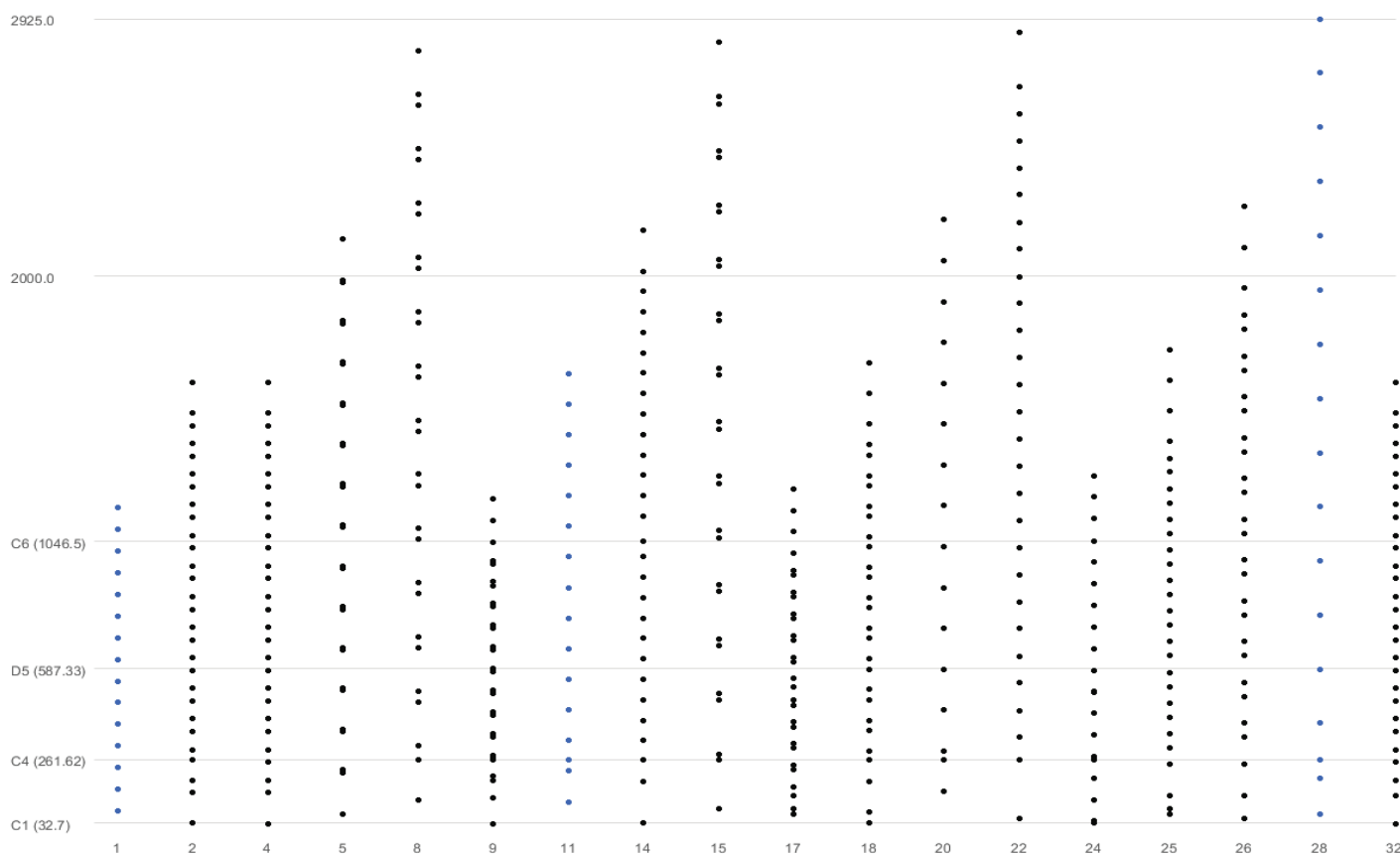


Fig. 33 - Les stack de fréquences de la section VI
 abscisse : événements ; ordonnée : fréquences

En abscisse se trouve non pas le temps mais les événements, à équidistance, et en ordonnée les fréquences des stack. En bleu sont indiqués les stack harmoniques. Cette représentation rend ainsi compte des deux écritures. Verticalement on voit les deux séries par stack de fréquences (on repère le même espacement par paires de points) ; horizontalement au niveau des fréquences maximales des stack on voit le schéma de quatre notes ascendantes se répéter quatre fois¹⁴³.

Précisons enfin que l'ensemble de ces écritures ne peut être audible puisqu'une enveloppe spectrale est appliquée sur les stack. Ces derniers ne sont donc jamais entendus tels quels, mais on les perçoit avec l'application de l'enveloppe, qui de plus se déplace continuellement avec le jeu de la flûte. On peut alors comprendre que l'enveloppe spectrale, en annulant des fréquences et en appliquant des formants, fait tendre ces stack vers le pôle de l'harmonicité.

¹⁴³ Le schéma est altéré à l'évt 4, où le stack de fréquence répète le précédent de façon percussive. Les événements 2 et 4 font partie en fait d'une même phrase comme nous le verrons dans l'analyse.

1.4 Phase Aligned Formant (Paf)

Ce module, certainement du fait qu'il n'était pas présent dans la première version de la pièce, se distingue très fortement des autres. Il utilise une méthode de synthèse éponyme, conçue en 1990 par Miller Puckette. Elle est présentée dans l'article [Puckette, 1996], puis dans le célèbre ouvrage [Puckette, 2006], où elle conclut le chapitre *design spectra*. Notons aussi qu'elle fait l'objet d'un brevet dont la propriété appartient à l'Ircam. Pour cette raison notamment (cf. le fichier *readme.txt* de l'archive), l'implémentation que nous avons lue varie de celle présentée dans [Puckette, 1996].

Enfin ce Module est spécifique puisqu'il est la réunion de techniques de synthèse, de traitement, et d'algorithmie¹⁴⁴. En effet le signal généré par la synthèse est suivi d'un traitement – qui est aussi utilisé pour un module *ad hoc* –, le Frequency Shifter, et des légers algorithmes sont intégrés pour faciliter son écriture.

1.4.1 Des formants

L'enjeu de la PAF est de permettre une description aisée du spectre avec des formants, et plus encore en décrivant une évolution dans le temps de ces formants. Pour rappel, les formants sont des pics d'énergie placés dans le spectre.

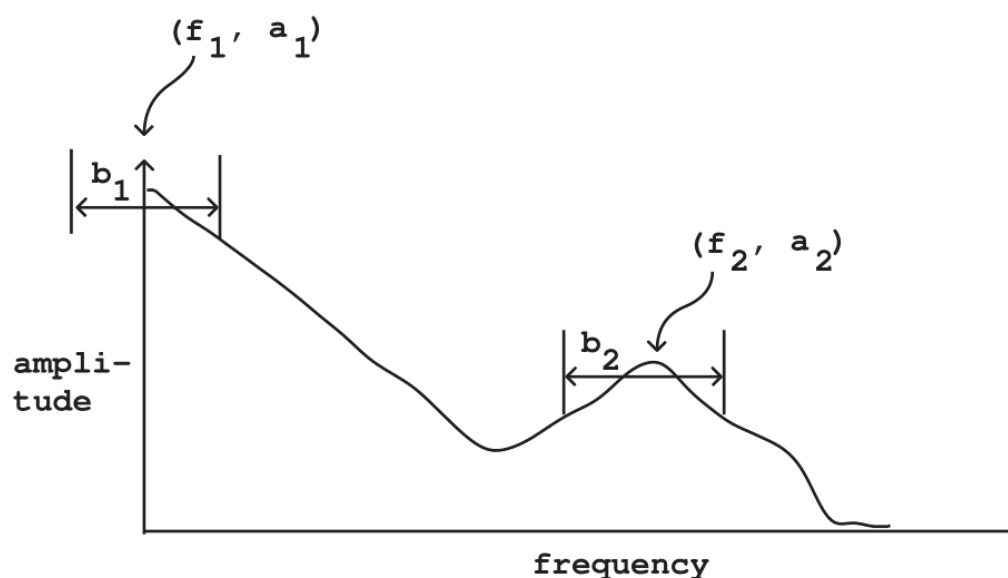


Fig. 34 - Description de formants, extrait de [Puckette, 2006, p. 148]
 f : fréquence, a : amplitude, b : largeur de bande

144 Il est ainsi une illustration du concept de Module que nous avons décrit au chapitre V.

Nous pouvons ainsi déduire trois paramètres de la PAF, relatifs à la description d'un formant : fréquence centrale (cf), largeur de bande (bw), et amplitude (amp).

1.4.2 La méthode de synthèse

L'idée maîtresse de la PAF est d'utiliser la modulation en anneau (ou ring modulation : RM par la suite) pour déplacer le long du spectre un formant déjà constitué. Le formant est alors le signal modulant (Mod) et il va être modulé par un autre, le porteur (Car). Avant la modulation, ce formant doit se situer dans le bas du spectre, de sorte que (i) son amplitude maximale soit atteinte pour la fréquence nulle (il s'agit donc d'un signal avec un DC offset), et (ii) l'amplitude des partiels décroissent proportionnellement avec la fréquence.

Pour envisager le signal résultant, il faut avoir en tête le principe de la RM : pour chaque composante sinusoïdale, la RM génère une paire de bandes latérales (sideband), situées de part et d'autre des fréquences du signal porteur. Par exemple, pour deux sinusoïdes modulées en anneau ayant pour fréquence 400 et 200Hz, le signal résultant est constitué de deux sinusoïdes situées à 200 et 600 Hz.

Le formant ayant sa plus forte amplitude pour la fréquence nulle, l'effet de la RM peut être vu comme l'application de ce formant (cette répartition d'harmoniques) à la fréquence spécifiée par le signal porteur.

Prenons un exemple :

- Soit le formant constitué des harmoniques avec les valeurs en fréquence suivantes :

$$ModFreqs=0, 100, 200, 300, 400, 500$$

- Soit le signal porteur avec la fréquence suivante :

$$CarFreq=500$$

- Pour chaque fréquence du formant une paire sera générée :

$$OutputFreq=(-500, 500), (-400, 600), (-300, 700), (-200, 800), (-100, 900), (0, 1k)$$

$$OutputFreq=0, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1k$$

Comme le signal modulant possède des amplitudes qui décroissent avec leurs fréquences, le signal résultant a bien l'allure d'un formant : l'harmonique avec la plus forte amplitude est la fréquence du signal modulant, et de part et d'autre de celle-ci, les autres amplitudes décroissent.

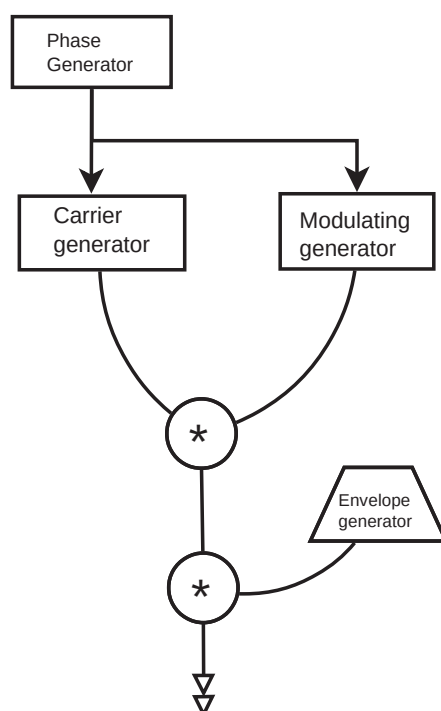


Fig. 35 - Schéma de synthèse général du Paf

La PAF se différencie de l'exemple simple que nous avons donné, car le modulant comme le porteur sont générés à partir de la même phase (phase aligned) : les perturbations de phase qui sont présentes en RM sont ainsi évitées, et la répartition en lobe de l'amplitude des harmoniques du formant est conservée.

1.4.3 L'implémentation

L'implémentation peut être décomposé en trois étapes. Dans la première il s'agit de générer le signal de phase à la fréquence fondamentale (la variable *pitch*). Dans la deuxième il s'agit de générer à partir de cette phase deux autres signaux, un modulant et un porteur. Dans la troisième il s'agit d'effectuer la modulation en anneau, et de contrôler l'amplitude du signal obtenu (la variable *amp*).

1.4.3.a Signal porteur

Le signal porteur est obtenu en multipliant le signal de phase par $cf/pitch$, puis en indexant une fonction gaussienne.

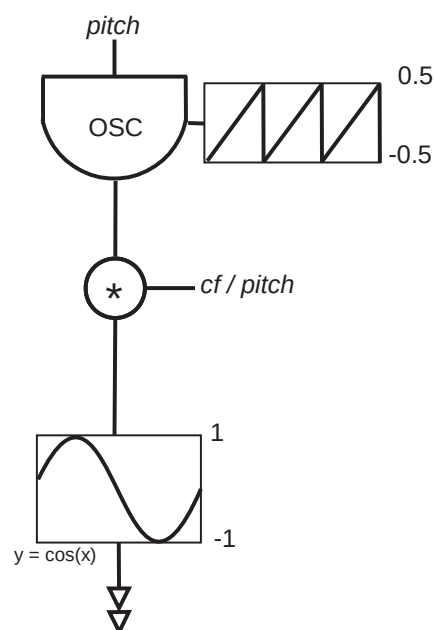


Fig. 36 - Génération du signal porteur

Il contient ainsi les harmoniques de rang impair de *pitch*. Leurs nombres et leurs amplitudes sont déterminés selon le rapport $cf/pitch$:

- S'il est entier, soit la fonction cosinoïdale oscille entre $-\pi$ et π , les amplitudes des partiels sont nulles et le signal est seulement composé de la fréquence fondamentale.
- S'il n'est pas entier la répartition des harmoniques impairs varie selon sa proximité avec un nombre entier. Par exemple une valeur de $cf/freq$ de 2.3^{145} , donnera le second harmonique avec une forte amplitude puis, celles qui suivent avec une amplitude nettement moindre. Une valeur de 2.7 donnera le plus d'énergie pour l'harmonique de rang 3, etc.

1.4.3.b Signal modulant

Il est obtenu par la synthèse *pulse train*, avec le paramètre *bandwidth* (*bw*) comme index de modulation. Ce dernier permet donc de contrôler le nombre d'harmoniques présents dans le formant.

145 Pour noter les décimales nous avons opté pour la notation anglo-saxonne.

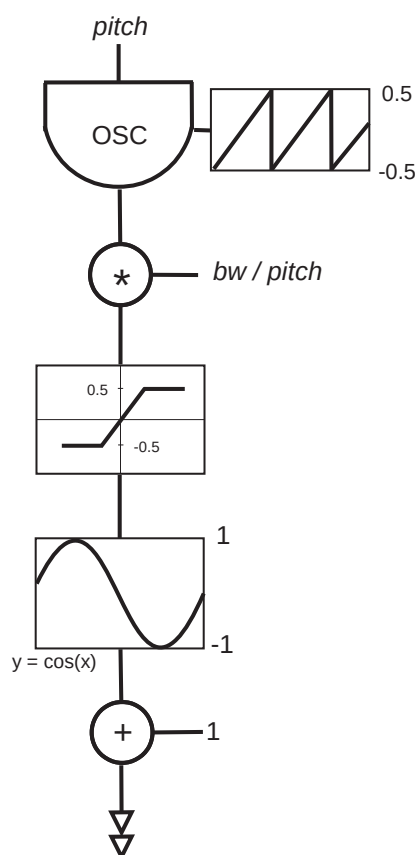


Fig. 37 - Génération du signal modulant

Le signal modulant contient la fréquence porteuse plus des harmoniques de rang impair avec la règle suivante : plus bw est conséquent, plus la forme d'onde est « pulsée », *i.e.* plus les harmoniques sont nombreux.

1.4.3.c Résultat

Le signal résultant est donc constitué d'un formant, dont la largeur est contrôlée par bw , et dont la position est contrôlée par cf . Le $pitch$ lui, détermine la valeur d'un fondamental fictif, et ainsi de l'écart en fréquence entre harmoniques.

La métaphore de la carte et du territoire peut ici être utile. Le $pitch$ trace une carte harmonique sur le territoire qu'est le spectre. Le formant va se positionner sur cette carte, en termes de fréquence centrale (cf), étendue (bw), et d'amplitude (amp).

La PAF est ainsi résolument harmonique, et s'il est question de formant, « d'énergie spectrale », ce n'est qu'à partir de la carte harmonique. Nous insistons car une lecture des écrits de la PAF nous laisse comprendre que son intérêt est de pouvoir placer des formants

indépendamment d'un fondamental. Or, comme nous l'avons vu, le positionnement des formants n'est pas à proprement parlé *indépendant* d'un fondamental, mais il se fait par une répartition en amplitudes des harmoniques de ce fondamental, comme si un *crossfade* s'opérait entre ces harmoniques :

The carrier signal that cross-fades between harmonics to make continuously variable center frequencies [Puckette, 2006, p. 167].

1.4.3.d Vibrato et Frequency Shifter

En plus de l'implémentation de la méthode PAF, à chaque voie est ajouté un vibrato et un Frequency Shifter (FS). Le premier, que l'on retrouve aussi dans *Additive*, permet d'ajouter de l'expressivité. Le second est un traitement que l'on retrouve dans un Module *per se* : il sera ainsi détaillé dans la suite de ce chapitre. Simplement, comme c'est la voie négative qui est effective, l'ajout du FS va permettre de décaler dans le grave toutes les composantes sinusoïdales. Les formants de la PAF vont ainsi pouvoir sortir de la carte harmonique tracée par le *pitch*.

1.4.4 Contrôle

Une abstraction, nommée *wavepacket*, permet de regrouper la production d'un formant avec le vibrato et le FS. Le Module est constitué de 8 de ces abstractions, placées en série et totalement indépendantes des unes des autres. Dans son ensemble, le Module peut donc engendrer un spectre de 8 formants, chacun indépendamment descriptible.

Chaque voie possède ainsi neuf paramètres. Tous sont suivis du numéro de la voie à laquelle ils se rattachent.

Nom de variable	Signification
pitch	Hauteur à percevoir du signal (MIDI)
freqshift	Valeur de décalage négatif (MIDI)
vamp	Profondeur du vibrato (amplitude linéaire, divisé par 512)
vfreq	Fréquence du vibrato (en Hz, divisé par 80)
amp	Amplitude du formant (MIDI)
noise	Envoi vers le module noise (MIDI)
cf	Center Frequency : fréquence centrale du formant (MIDI)
cf-fline	Temps de glissement pour cf (ms)
bw	Bandwidth : largeur de bande (MIDI)

Tab. 10 - Contrôle pour une voie du Module Paf

1.4.5 Connexion au Noise

La connexion au routing interne est très particulière puisque chacune des voies est reliée au Module *Noise*, par une variable éponyme suivie du numéro de la voie. C'est le seul Module à présenter de telles *connexions figées* à l'échelle des voies, et non du Module. En suivant la logique de l'implémentation, la connexion du Paf – comprendre « du Module Paf » – au Noise devrait se faire à l'échelle du groupement de Modules, nommé Osc, et donc avec la variable *oton*, bien présente.

Ces particularités d'implémentation – que l'on pourrait voir comme des incohérences dans la logique de réalisation de l'implémentation – viennent compliquer le travail sur le routing que nous présentons au chapitre suivant.

1.4.6 Algorithmes

Le Module possède deux légers algorithmes. Le premier permet de dessiner des enveloppes d'amplitude avec des suites de points. Avec l'implémentation actuelle le contrôle de la Paf se fait « dans le temps » à l'instar d'un fader sur une table de mixage. Ce premier léger algorithme permet au compositeur de prescrire des enveloppes d'amplitude à l'aide de paires de données (valeur de destination, temps en milliseconde). Par exemple la ligne suivante

```
env1 -1, 150 5, 120 900, -2;
```

dessine un début d'enveloppe avec une attaque (150 5) et un déclin (120 900)¹⁴⁶. Au maximum l'enveloppe d'amplitude peut contenir douze points successifs. Comme le montre cet exemple, ces nouveaux contrôles se font à l'échelle d'une voie, à l'aide de variables commençant par *env*.

Le second algorithme, lui, s'établit au niveau du Module. Il vient faire face au problème posé par le trop grand nombre de contrôles : 8 voies à décrire indépendamment avec 9 variables. La solution réalisée est d'implémenter un *arpeggiator* qui va distribuer des valeurs aux variables communes de ces huit voies. Par exemple la ligne suivante

```
pufbw -1, 0, 0, 0, 0, 45, 50, 48, 40;
```

distribue aux huit variables *bw*[1-8] respectivement les valeurs 0, 0, 0, 0, 45, 50, 48 40¹⁴⁷.

Relevons que l'implémentation réalisée dépasse l'usage qui en est fait. Effectivement, le propre d'un *arpeggiator* est de distribuer *dans le temps* des valeurs. Or, nous l'avons découvert après des lectures automatiques des qlist (cf. Chapitre VIII 1), les valeurs de temps ne sont jamais

¹⁴⁶ Les données -1 et -2 sont des conventions fixées par le designer ; elles indiquent le début et la fin de la description d'une enveloppe.

¹⁴⁷ C'est donc l'index des données qui détermine le numéro de la voie. La donnée -1 est une convention du designer, elle remet à zéro l'index.

utilisées, ou pire, elles le sont mais avec des valeurs nulles – ce qui vient considérablement alourdir la lecture de ces « partitions » alphanumériques. L'algorithme permet donc simplement d'affecter des valeurs aux variables communes des voies. Il est utilisé pour le *pitch*; pour la description des formants (*cf*, *amp* et *bw*); et pour le vibrato (*vamp* et *vfreq*). Six contrôles viennent donc s'ajouter, commençant par *puf*.

Nom de variable	Signification
<i>pufpitch</i>	Distribue ses valeurs aux variables <i>pitch</i> des 8 voies
<i>pufcf</i>	Distribue ses valeurs aux variables <i>cf</i> des 8 voies
<i>pufbw</i>	Distribue ses valeurs aux variables <i>bw</i> des 8 voies
<i>pufamp</i>	Distribue ses valeurs aux variables <i>amp</i> des 8 voies
<i>pufvfreq</i>	Distribue ses valeurs aux variables <i>vfreq</i> des 8 voies
<i>pufvamp</i>	Distribue ses valeurs aux variables <i>vamp</i> des 8 voies

Tab. 11 - Contrôle du Paf¹⁴⁸

1.4.7 Utilisations

Comme nous l'avons vu au chapitre précédent le Paf¹⁴⁹ est particulier puisque c'est un Module qui a été implémenté après la création de la pièce. Son intégration ne va donc pas changer la structuration de la pièce, mais elle s'est faite avec un travail d'orchestration ([Manoury, 2017]). L'usage principal de ce Module consiste ainsi à enrichir des agrégats déjà existants, produits généralement par la flûte et l'Harmonizer, et maintenus par la Reverb. Par exemple, le fondamental d'une note jouée par la flûte est repris dans la variable *pitch* et, avec *cf*, un formant va être placé 6 demi-tons au-dessus de *pitch*. La deuxième voie pourra être utilisée similairement avec un deuxième formant placé 8 demi-tons au-dessus¹⁵⁰. Les autres voies peuvent être utilisées avec des valeurs de *pitch* différentes de sorte à enrichir l'agrégat.

On trouvera deux autres usages, plus saillants, dans les trois premières sections. Le rendu se démarque davantage puisque le Paf produit seul les agrégats, avec un rythme plus soutenu (section III), ou encore très localement dans les deux premières sections où des *éclats* sont produits avec des formants percussifs placés dans le sur-aigu (e.g. section I, evt 8).

¹⁴⁸ Trois de ces variables ont été renommées pour plus de cohérence : *pufpitch* remplace *puffq*, *pufvfreq* remplace *vibfreq*, et *pufvamp* remplace *vibdepth*.

¹⁴⁹ Comprendre « le Module Paf », et non « la méthode de synthèse PAF »

¹⁵⁰ Le début de la sous-section IA fonctionne ainsi.

2 Modules de traitement

Nous décrivons ici les quatre Modules de traitement, Harmonizer, Frequency Shifter, Reverb et Noise, avec la même grille que précédemment. La troisième étape de cette grille, celle de l'utilisation, est ici importante puisque nous nous sommes aperçus que ces Modules possédaient des configurations types, rappelées au cours de la composition.

En préambule précisons que ces modules de traitement sont appliqués sur des signaux qui sont déjà diffusés. Il ne s'agit donc pas de créer un signal avec de nouvelles significations, mais plutôt de modifier un signal déjà porteur de significations, de sorte à l'enrichir.

2.1 Harmonizer

Comme l'explique le compositeur ce Module « permet de transposer un son dans l'aigu ou dans le grave sans en changer la durée ». Il est de plus « associé à un delay permettant de produire la transposition voulue avec un retard calculé ». [Manoury *et al.*, 2003, *Jupiter*]. Cet Harmonizer est fait de quatre voies, chacune contrôlée avec 3 variables, *hfre*, *hdel* et *hamp*, et une autre, commune aux quatre voies, *hwind*.

2.1.1 Description d'une voie

Le cœur de l'Harmonizer est une ligne de délai qui enregistre le signal entrant sur 5 secondes. Cette ligne est ensuite lue dans chacune des voies, et ceci à deux reprises, avec des vitesses de lecture variables. Nous décrivons ci-après le procédé pour une seule des voies.

La transposition se fait en lisant la ligne de délai avec une vitesse variable : le temps de retard avec lequel la ligne doit être lue est prescrit avec un signal en dents de scie. La ligne n'est pas lue dans son entièreté, mais selon une fenêtre temporelle dont la durée est contrôlée avec la variable *hwind*. Le taux de transposition est indiqué par *hfre* et c'est le rapport $hfre/hwind$ qui détermine la fréquence du signal en dents de scie.

La possibilité d'appliquer un temps de retard avant la lecture du délai se fait simplement en ajoutant une constante avant la lecture de la ligne de délai, le paramètre *hdel*. Afin d'éviter des clics audios lors des enchaînements des fenêtres audio une enveloppe d'amplitude cosinusoidale est appliquée à l'issue de la lecture de la ligne de délai.

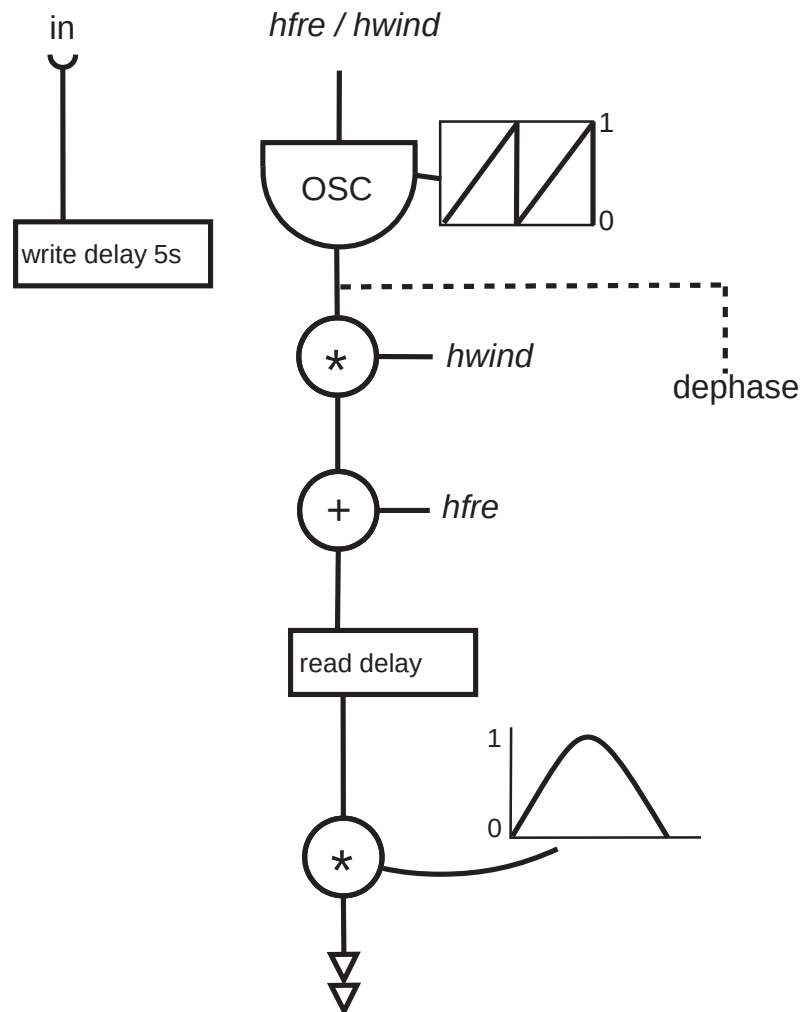


Fig. 38 - Schéma de traitement d'une voie de l'Harmonizer

Pour gagner en qualité de traitement ce procédé de transposition est dupliqué pour chaque voie : le signal de phase en dents de scie est déphasé de $\pi/2$ de sorte que lorsque la première lecture de la ligne de délai se trouve au tout premier échantillon, la seconde se trouve, elle, à sa moitié, et ainsi de suite.

Enfin, après sommation des deux signaux, une enveloppe d'amplitude est appliquée. Elle permet d'éviter les clics audios qui pourraient apparaître lors des changements de valeurs des paramètres. Elle s'enclenche ainsi automatiquement pour chaque changement de *hfre* ou *hdel*.

2.1.2 Contrôle

Ce que nous avons décrit est regroupé au sein d'une abstraction, nommée *jup-harm*, et le Module en est composé de quatre, autorisant ainsi 4 transpositions simultanées.

En plus des trois contrôles clés *hfre*, *hdel* et *hamp* un autre est implémenté au-dessus de ces derniers : *trans* affecte, avec la même valeur, *hfre* et affecte *hamp* d'une valeur de 127. Il permet

donc une légère économie d'écriture en spécifiant la valeur de transposition sans se soucier de l'amplitude de la voie. Les trois contrôles clés sont présents dans chacune des voies, suivis du numéro de la voie auquel ils sont attachés (e.g. *hdel3*). En plus de ces quatre contrôles s'ajoute *hwind*, commune à toutes les voies, qui permet de contrôler la qualité de la transposition.

Nom de variable	Signification
<i>hfre</i> [1-4] ¹⁵¹	Fréquence de transposition (MIDI cents)
<i>hdel</i> [1-4]	Délai à appliquer (ms)
<i>hamp</i> [1-4]	Amplitude de la voie (MIDI)
<i>trans</i> [1-4]	Affecte <i>hamp</i> = 127, et <i>hfre</i> = <i>trans</i>
<i>hwind</i>	Taille de la fenêtre en MIDI (127 = 4s)

Tab. 12 - Contrôle de l'Harmonizer

2.1.3 Utilisations

Deux utilisations bien distinctes de l'Harmonizer sont repérables. La première est standard, elle permet de transposer le signal entrant dans l'espace des hauteurs. Ce procédé étant augmenté par le délai, le placement de sons peut se faire et dans l'espace des hauteurs et dans le temps. Ainsi le module peut prendre une grande importance, en pouvant « jouer » à n'importe quel moment, en dehors du jeu de la flûte, ce qui autorise une écriture en contrepoint. Par cette indépendance, le Module, bien qu'il implémente une méthode de traitement, s'approche du type *synthèse*.

La seconde utilisation est nettement moins signifiante musicalement, elle consiste à venir flouter le signal entrant – Nous parlons ainsi du *mode flouteur* de l'Harmonizer. On trouve en effet une utilisation avec des faibles valeurs pour les *hfre*, moins de 1/4 de demi-ton¹⁵². Il ne s'agit plus ici de transposer un signal mais de l'enrichir, de lui apporter de la consistance en lui ajoutant quatre signaux quasi identiques¹⁵³.

¹⁵¹ Pour alléger notre écrit nous écrivons *hdel*[1-4] pour désigner les quatre contrôles *hdel* du Module, et de même pour les autres contrôles qui se démultiplient sur les voies.

¹⁵² C'est une configuration récurrente avec les mêmes valeurs : -18, -6, 16, 17 centièmes de note MIDI.

¹⁵³ Ce que réalise déjà le module figé Phase Modulation, avec deux voies, ancré sur la flûte (cf. chapVI 2.3).

2.2 Frequency Shifter

Le *Frequency Shifter* (FS par la suite) permet de décaler en fréquence les composantes sinusoïdales d'un signal. Le Module se contrôle avec trois variables, *fsfre*, *fpos* et *fneg*, et par la nature même du traitement il est fait de deux voies, une nommée positive et l'autre négative. Comme précédemment il y a deux usages du Module.

2.2.1 Description

Ce Module peut paraître étonnant dans la mesure où il mobilise très peu de boîtes objets mais qu'il reste néanmoins le traitement le plus opaque. La raison de ceci est que le cœur du traitement est une transformation de Hilbert (une combinaison de filtre passe-tout), et que la compréhension de cette transformation appelle des connaissances conséquentes en ingénierie audio (cf. [Baudry *et al.*, 2015] pour une présentation de différentes implémentations et pour un aperçu de son usage en musique mixte). Ainsi l'ouvrage de [Puckette, 2006, p. 259], pourtant proche du domaine de l'ingénierie, renvoie aux sources explicatives. Il est donc d'usage de décrire et de comprendre le *Frequency Shifter* en suivant le principe de la boîte noire, en décrivant ce qu'il fait et non comment cela est obtenu.

Nous nous basons sur l'explication présentée par Philippe Manoury dans [Manoury *et al.*, 2003, *Jupiter*]. Le signal entrant est considéré comme un ensemble de composantes sinusoïdales, ayant pour fréquence $f_1, f_2, f_3 \dots f_n$. À partir d'une sinusoïde contrôlée en fréquence, la variable *fsfre*, le traitement calcule deux autres signaux ayant le même nombre de composantes sinusoïdales, avec les mêmes amplitudes, mais dont les fréquences changent de la façon suivante :

- le premier est constitué des fréquences $f_1 - fsfre, f_2 - fsfre, f_3 - fsfre \dots f_n - fsfre$
- le second des fréquences $f_1 + fsfre, f_2 + fsfre, f_3 + fsfre \dots f_n + fsfre$

Ces deux signaux sont ensuite contrôlés en amplitude. Le premier par la variable *fneg*, le second par *fpos*. Le fait que l'on puisse contrôler indépendamment « la soustraction » et « la sommation » est ce qui différencie le *Frequency Shifter* du modulateur en anneau (RM), pour lequel ces deux constituants ont nécessairement la même amplitude, comme le précise P. Manoury :

Un des problèmes du modulateur en anneaux est sa forte densité spectrale qui, lorsque le son modulé est déjà très riche en contenu harmonique, parvient assez vite à créer une sensation de saturation dans laquelle il est parfois difficile d'identifier des « couleurs » inharmoniques particulières. Lorsque l'on sépare ces deux fonctions, on obtient des spectres inharmoniques qui ne possèdent que la moitié des composants que ceux que

fournit la modulation en anneau (P. Manoury, Traité d'inharmoine, philippemanoury.com/?p=4965, 2017-08-02).

2.2.2 Contrôle

Nom de variable	Signification
<i>fpos</i>	Valeur de décalage en fréquence (MIDI)
<i>fneg</i>	Amplitude de la voie négative (MIDI)
<i>fpos</i>	Amplitude de la voie positive (MIDI)

Tab.13 - Contrôle du Frequency Shifter

2.2.3 Utilisations

Par sa proximité avec la Ring Modulation le FS possède au moins deux utilisations. Une première avec soit *fpos* soit *fneg* actif, et une seconde avec $fpos = fneg$. La première correspond à un décalage fréquentiel des composantes ; la seconde à l'effet d'un modulateur en anneau¹⁵⁴.

Comme nous l'annonçait la précédente citation du compositeur, le mode RM, afin d'éviter la « sensation de saturation », est utilisé avec des valeurs de *fsfre* situées dans l'infra audio. En doublant chaque composante avec une faible valeur, cette utilisation engendre l'effet connu de battements, ce qui va permettre d'ajouter du relief momentanément sur un signal.

Il est important de relever que ce traitement s'applique en parallèle d'une diffusion du signal source (*dry*). Dès lors, quelle que soit l'utilisation qui en est faite, le FS vient brouiller un signal déjà porteur de sens.

2.3 Réverb

Le Module Reverb¹⁵⁵ permet simplement de prolonger le son dans le temps. Il a cependant un usage particulier puisque parfois ce prolongement peut se faire indéfiniment. Trois paramètres sont sollicités *rgate*, *rout* et *revfb*.

2.3.1 Description

Le traitement est divisé en deux blocs. Dans le premier il s'agit d'épaissir le signal à l'aide d'une combinaison de filtres en peigne et d'une sommations et de soustractions de signaux ; dans le second il s'agit de *prolonger* le signal à l'aide d'une matrice circulaire.

¹⁵⁴ Nous parlerons de mode RM du Frequency Shifter.

¹⁵⁵ Afin d'alléger nous utilisons cette forme courte, plutôt que *réverbération*.

2.3.1.a Premier bloc : épaissir le signal

Ce bloc est fait de 5 étapes successives, toutes identiques, qui permettent d'épaissir le signal. Nous proposons de le décrire ainsi :

Soit un signal entrant et son double retardé de 75 ms. La première étape consiste à (i) faire la somme de ces signaux (le résultat est nommé *signal 1a*) et (ii) d'enregistrer la différence de ces signaux dans une nouvelle ligne de délai de 43 ms (le résultat est nommé *signal 1b*).

De même, la seconde étape consiste à (i) faire la somme des signaux ($1a + 1b = 2a$) et (ii) d'enregistrer la différence des signaux ($1a - 1b = 2b$) dans une nouvelle ligne de délai de 25 ms. Et ainsi de suite avec des valeurs de délai dégressives, jusqu'à la sixième et dernière étape qui reçoit les signaux 5a et 5b, et qui crée les signaux 6a et 6b.

Parallèlement nous pouvons décrire ce procédé de façon algébrique, de la façon suivante¹⁵⁶ :

Soit *in* un signal donné, et la notation $del(in, 75)$ un retard de 75 ms appliqué à *in*.

$$1a = in$$

$$1b = del(in, 75)$$

$$2a = 1a + 1b$$

$$2b = del(1a - 1b, 43)$$

$$3a = 2a + 2b$$

$$3b = del(2a - 2b, 25)$$

$$4a = 3a + 3b$$

$$4b = del(3a - 3b, 19)$$

$$5a = 4a + 4b$$

$$5b = del(4a - 4b, 16)$$

$$6a = 5a + 5b$$

$$6b = del(5a - 5b, 13)$$

Enfin, pour illustrer ce premier bloc, on peut s'imaginer une salle virtuelle et un signal entrant fait d'une impulsion. Ce signal viendrait alors rebondir contre les parois de la salle virtuelle, en de nombreux « points d'écho ». Alors, comme le souligne M. Puckette, chacune des 6 étapes de ce premier bloc revient à doubler le nombre d'impulsions : pour une seule impulsion dans le signal source la salle virtuelle crée un nouveau signal constitué de 64 impulsions (2^6) cf. [Puckette, 2006, p. 193].

¹⁵⁶ Les temps de délai ont été arrondis

2.3.1.b Second bloc : la réverbération

Le second bloc insère les deux signaux du précédent bloc (6a et 6b) dans une matrice circulatoire faite de quatre lignes de délai. La matrice est dite circulatoire puisque d'une part deux boucles de rétroaction (*feedback*) sont présentes : la lecture des lignes de délai est ré-enregistrée dans ces mêmes lignes de délai. C'est ici qu'intervient le contrôle le plus important du Module, *revfb*, qui va contrôler la quantité de signal à réinjecter dans cette matrice. Et d'autre part, une matrice de sommation et soustraction est réalisée. Comme précédemment nous pouvons décrire cette matrice avec une vue algébrique. Considérons :

- les quatre lignes de délai r, s, t et u ;
- les temps de retard de ces lignes sont fixes, ils valent respectivement 58.6435, 69.4325, 74.5234 et 86.1244 ms ;
- la notation $fb(r)$ indiquant une lecture de la ligne de délai r avec un taux d'atténuation correspondant à *revfb*.

La matrice circulatoire est alors la suivante :

$$\begin{aligned} r &= fb(r) + 6a + fb(s) + 6b + fb(t) + fb(u) \\ s &= fb(s) + 6b - \{ fb(r)+6a \} + \{ fb(t) - fb(u) \} \\ t &= fb(t) + fb(t) - \{ fb(r) + 6a + fb(s) + 6b \} \\ u &= fb(t) - fb(u) - \{ fb(r) + 6a - (fb(s) + 6b) \} \end{aligned}$$

On lit ici la circularité : chacun des enregistrements des lignes de délai est lié aux lectures des autres lignes de délais.

2.3.2 Contrôle

La Reverb possède trois contrôles. Le premier essentiel est *revfb*. En modifiant la quantité de signal à réinjecter dans les quatre lignes de délai de la matrice circulatoire, *revfb* permet simplement de modifier la durée de la réverbération. Les deux autres sont des paramètres d'amplitude.

Nom de variable	Signification
revfb	Durée de la réverbération (MIDI)
rgate	Amplitude du signal entrant
rou	Amplitude du signal produit

Tab.14 - Contrôle de Reverb

2.3.3 Sortie

Le Module possède deux sorties, prélevées au croisement des deux blocs du Module. À la suite de notre explication algébrique ils valent $6a+fb(r)$ et $6b+fb(s)$. Ce sont ces deux signaux qui sont adressés à la diffusion stéréophonique. Pour le routing interne, à la différence de tout les autres Modules, ce n'est pas la sommation qui est utilisée mais seulement le premier de ces signaux.

2.3.4 Utilisations

Selon la valeur de *revfb* nous pouvons différencier trois utilisations

- $revfb = 0$

Alors la matrice circulatoire n'est pas effective et l'on entend le résultat du premier bloc du traitement : le signal entrant est épaissi avec 64 points d'écho. Cet usage reste très peu fréquent, on le trouve par exemple dans la section XII à l'événement 14.

- $0 < revfb < 100$

Ici la réverbération est dans un usage classique : elle prolonge le signal entrant proportionnellement à *revfb*. C'est un usage qui relève du mixage ; il est présent par exemple dans la section VI.

- $revfb = 127$

Il s'agit d'une utilisation récurrente, chère au compositeur, qui permet de prolonger indéfiniment le signal entrant. Cela permet de former et de prolonger des agrégats, par exemple trois notes de flûte prolongées en parallèle.

Il est possible ainsi de créer des « rémanences sonores » par l'entretien de certaines notes d'une mélodie. La réverbération infinie agit ici comme une pédale de piano « sélective » qui ne prendrait que certaines notes d'un fragment et non sa totalité [Manoury *et al.*, 2003, *Jupiter*].

C'est dans ce cas de figure que *rgate* prend tout son sens : il permet de sélectionner les notes à prolonger, ou réciproquement de sélectionner les notes à ignorer¹⁵⁷.

¹⁵⁷ Le compositeur illustre ce procédé avec une métaphore : [ce système permet de capturer des notes avec une grande précision] un peu comme le ferait une chauve-souris avec des insectes ! *Ibidem*.

2.4 Noise

C'est un traitement qui permet de bruiteur un signal. Il se contrôle avec une seule variable, *fnois*, et à recours à une forme d'onde, *noise.snd*. De tous les traitements c'est celui qui a la moindre importance musicale ; il est ainsi absent de la documentation CdRom [Manoury *et al.*, 2003].

2.4.1 Description globale

Le Module utilise à deux reprises la synthèse par modulation d'anneau (RM). Une première fois pour générer, avec la forme d'onde *noise*, un signal modulant, riche, fait de beaucoup de variations d'amplitude, et une seconde fois pour moduler le signal entrant avec ce précédent signal. Le paramètre *fnois* impacte le signal modulant, il détermine la vitesse de lecture de la forme d'onde *noise*.

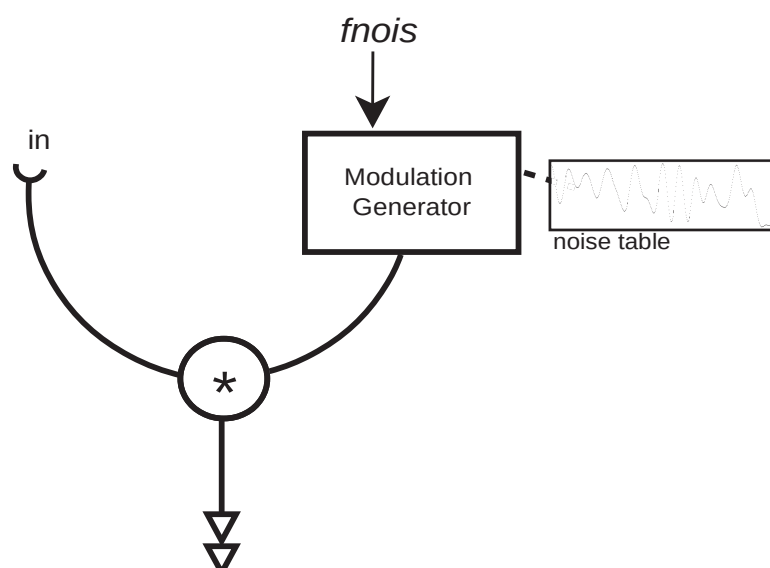


Fig. 39 - Schéma de traitement global du Noise

2.4.2 Description détaillée

À des fins de qualité audio ce principe simple de modulation est effectué à quatre reprises avec des signaux proches mais non identiques.

C'est d'abord le signal entrant qui est démultiplié à quatre reprises à l'aide de filtres en peigne (quatre lignes de délai de moins de 25 ms). Ce sont ensuite les signaux modulants qui sont démultipliés avec une matrice que nous détaillons ci-après.

Matrice des signaux modulants

Les signaux modulants sont construits en deux étapes. La première consiste à générer 8 signaux avec des oscillateurs lisant la forme d'onde *noise*. Ces oscillateurs ont tous des valeurs en fréquence différentes mais avoisinantes, elles sont déduites de *fnois* par des coefficients multiplicateurs compris entre [0.22 ; 0.299]. Ensuite, une synthèse RM est effectuée entre eux de sorte à obtenir quatre nouveaux signaux.

La seconde étape consiste elle à enrichir ces quatre signaux à l'aide d'une matrice de soustraction et sommation comme nous avons pu en trouver dans la Reverb.

Soit i_1, i_2, i_3 et i_4 les signaux entrant dans la matrice, et o_1, o_2, o_3 et o_4 les signaux en sortant.

$$\begin{aligned} o_1 &= (i_1+i_2) + (i_3+i_4) \\ o_2 &= (i_1+i_2) - (i_3+i_4) \\ o_3 &= (i_1-i_2) + (i_3-i_4) \\ o_4 &= (i_1-i_2) - (i_3-i_4) \end{aligned}$$

o_1, o_2, o_3 et o_4 sont donc les quatre signaux modulant qui vont moduler les quatre signaux du signal entrant, obtenus avec les filtres en peigne.

2.4.3 Contrôle

Le contrôle du Module se fait donc uniquement avec la variable *fnois*. L'implémentation est telle que *fnois* permet d'avoir un contrôle sur la quantité de modulation, il devient ainsi un *index de modulation*, et ceci indépendamment de l'amplitude du signal de sortie – ce qui n'est pas le cas dans une RM simple. Ainsi, plus *fnois* est grand plus le bruit est présent en sortie, jusqu'à ce qu'il soit le seul présent.

2.4.4 Utilisations

Deux utilisations peuvent être distinguées selon *fnois*.

- $fnois < 70$

Il s'agit de l'utilisation usuelle. Le signal modulant est alors dans l'infra audio et le traitement revient alors à appliquer une sorte d'enveloppe d'amplitude au signal entrant, mais cette enveloppe est résolument complexe car non cyclique (du moins à la perception), et avec des inversions de phases.

La complexité du traitement vient de la combinaison des trois points suivants : (i) la démultiplication du signal entrant avec les 4 filtres en peigne, (ii) les différents coefficients multiplicateurs après la réception de *fnois*, et (iii) la matrice de soustraction sommation pour les signaux modulants.

En écoute de laboratoire¹⁵⁸, lorsque ce Module est directement affecté sur la flûte, et sans diffuser le son source, on reconnaît bien l'instrument, mais le son est haché et sans que l'on puisse repérer un quelconque *pattern*.

- $fnois > 70$

Le signal modulant est maintenant dans le domaine audible (~6 hz pour la lecture des tables) : le traitement n'applique pas de bruit sur le signal mais effectue de la synthèse. Dès lors qu'il y a du signal en entrée la RM va engendrer un signal très riche constitué de nombreux partiels, fixes dans la répartition, mais tellement nombreux que l'on perçoit du bruit situé dans le spectre. Le bruit perçu est d'autant plus élevé que *fnois* est conséquent.

Cette utilisation « en synthèse » est présente une seule fois, dans la section X (evt 2 - 10) où *fnois* vaut 108. La flûte possède alors un double électronique, qui est produit par l'Harmonizer suivi du Noise.

2.5 Spat

Comme nous l'avons vu le Spat se démarque des autres Modules puisqu'il ne peut pas être saisi à partir d'une compréhension « centrée sur le signal » : il est en effet nécessaire de prendre en compte la disposition des haut-parleurs, la diffusion. De ce fait c'est un Module qui déborde de la classification synthèse ~ traitement – nous l'intégrons néanmoins à la suite des traitements par proximité. Ce Spat peut être compris comme l'assemblage de trois éléments¹⁵⁹. Il y a d'abord la simulation d'un espace bidimensionnel, à l'aide du *quadra* ; il y a ensuite au sein de cet espace la description de quatre déplacements ; et enfin il y a l'intégration d'une réverbération qui permet de doser la présence de ces déplacements.

2.5.1 Simuler un espace bidimensionnel

Au centre de ce Module il y a la simulation d'un espace de diffusion à l'aide des quatre haut-parleurs du *quadra*. Cet espace, rectangulaire, est investi avec deux dimensions x et y, longueur et largeur. La difficulté de l'implémentation réside alors dans la correspondance à effectuer entre le positionnement d'un son dans cet espace et le niveau d'amplitude à attribuer aux 4 haut-parleurs. Par exemple, quel dosage des 4 signaux permet de positionner un son dans la partie haute droite (x = 75 % et y = 10%) ?

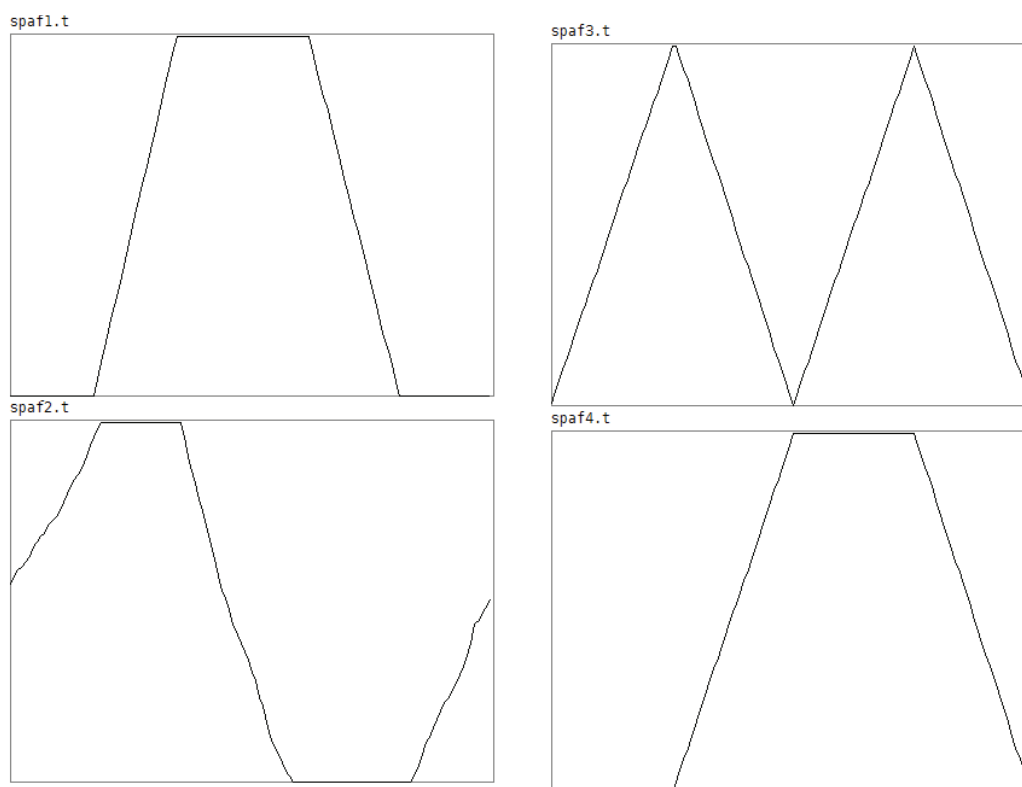
¹⁵⁸ C'est-à-dire que le Module est recopié dans un nouveau patch et l'on écoute le résultat du traitement avec différents paramétrages.

¹⁵⁹ En raison de notre positionnement théorique (cf. Chapitre IV 2.3.1), qui ne se base pas sur la diffusion des signaux, la description que nous faisons du Spat ne respecte pas la grille usuelle.

Ceci est réalisé par un calcul mathématique que nous ne détaillerons pas. Précisons simplement que les deux dimensions sont calibrées selon l'échelle MIDI, de 0 à 127, et que l'implémentation intègre un contrôle sur la taille de l'espace à simuler, la variable *radius*. Cette dernière permet de doser la taille de ces dimensions, soit d'avoir la main sur l'importance des déplacements à effectuer.

2.5.2 Quatre déplacements possibles

Au sein de cet espace quatre déplacements sont possibles. Chacun de ceux-ci est décrit avec deux tables de données, une par dimension, de 127 index. La figure ci-dessous présente les deux premiers déplacements de gauche à droite, avec la dimension x en haut.



*Fig. 40 - Représentation de deux déplacements.
Un déplacement est fait de la combinaison de deux tables verticales.
En haut les tables pour x ; en bas les tables pour y.*

Au début du premier déplacement par exemple (tables sur la gauche), le son se trouve à 0 % de x et 50 % de y, puis à la moitié de la table, le son va se déplacer jusqu'à x = 100 % et y = 50 %.

La lecture de ces tables se fait en boucle, après sélection et déclenchement avec la variable *spaton* : lorsque cette variable vaut 2 c'est le deuxième mouvement qui est lu, etc. La lecture des tables se fait avec un métronome de 10 ms et une variable *ad hoc*, *spatinc*, qui détermine le saut

d'index à effectuer. *spatinc* permet donc de contrôler la vitesse des déplacements spatialisés¹⁶⁰. De plus elle peut obéir à des temps de lissage, ce qui permet par exemple de passer d'un déplacement très vif à un autre très lent. Ce procédé est très utilisé par le compositeur – *spatinc* est la variable la plus utilisée du Spat –, cela permet par exemple de faire des *rallentando*, très saillants, qui viennent généralement marquer la fin de phrases (*e.g.* dans la section IV).

2.5.3 La réverbération

En amont de ces déplacements un traitement de réverbération est appliqué au signal entrant. Il s'agit de la même implémentation que celle du Module de Reverb : d'abord une série de 6 filtres en peigne et ensuite une matrice circulatoire avec quatre lignes de délais. Nous retrouvons ainsi les mêmes paramètres : *zdist* la quantité de réverb à appliquer (dry/wet) ; *zgate* le niveau de signal à enregistrer dans les délais ; *zevfb*, la durée de la réverbération simulée et enfin *zout* le niveau de sortie des signaux réverbérés.

2.5.4 Contrôle

Le Spat possède donc 7 contrôles, 3 relatifs aux déplacements à effectuer et 4 pour paramétrer la réverbération.

Nom de variable	Signification
radius	Profondeur de l'espace simulé
spatinc	Vitesse de lecture des tables
spaton	Sélection et lecture en boucle d'un déplacement
zdist	Quantité de réverbération à appliquer
zevfb	Longueur de la réverbération à simuler
zgate	Niveau de signal entrant à enregistrer
zout	Niveau de réverbération en sortie

Tab. 15 - Contrôle du Spat

¹⁶⁰ Nous relevons que ces mouvements de spatialisation possèdent leurs propres temporalités, qui n'est ni celle du flûtiste ni celle en millisecondes des séquences du Sampler.

3 Structure de contrôle

La structure de contrôle de *Jupiter* se dessine au travers des nombreux contrôles que nous avons abordés dans ce chapitre. En mettant de côté les démultiplications des contrôles sur les nombreuses voies que peut avoir un Module (e.g. *hfre1*, *hfre2* ; *pitch4*, *pitch5*, etc.), on peut distinguer environ 25 contrôles. Évidemment tous n'ont pas le même poids de signification. Par exemple un contrôle comme *chapo-tab* du *Chapo* est moins signifiant, porte moins de signification, que le contrôle *cf* (center frequency) du *Paf*. De même, un contrôle d'amplitude, s'il est utilisé normalement, apparaît peu porteur de signification.

Afin de mettre en lumière la structure de contrôle il semble alors possible de différencier parmi tous ces contrôles des poids de significations. C'est ce que nous avons réalisé dans la figure ci-dessous, en mettant de côté les contrôles de « bas niveau » comme ceux qui concernent l'amplitude donc, et en ne tenant pas compte des voies des Modules.

Il faut préciser avant tout que l'intérêt d'un tel schéma ne se trouve pas à l'échelle d'un contrôle, ou d'un Module, mais bien plus à une échelle globale, qui regroupe tous les contrôles et qui permet ainsi de les mettre en rapport.

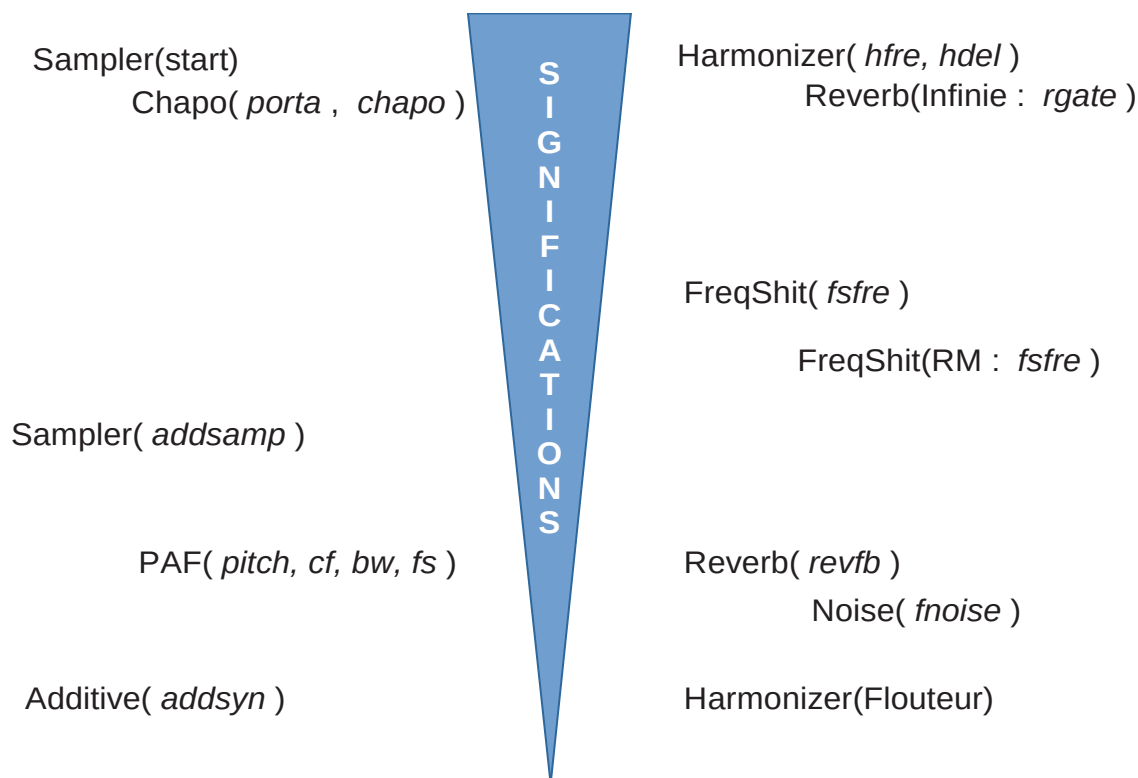


Fig. 41 - Les significations présentes dans la structure de contrôle.
Les voies des Modules et les contrôles d'amplitude sont ignorés.

Les Modules de synthèse sont situés sur la droite, et ceux de traitement sur la gauche. En effet, puisque les Modules de traitement ne peuvent être signifiants qu'en connaissant le signal entrant, il nous a paru difficile d'envisager la signification de ces derniers conjointement à ceux de synthèse.

Les Modules de traitement qui ont plusieurs modes sont présents plusieurs fois¹⁶¹. Ainsi de l'Harmonizer, très signifiant quand il permet de transposer et de retarder un signal, et résolument moins quand il est utilisé en « mode flouteur ».

Les contrôles du Sampler (en mode séquence) et du Chapo, du fait qu'ils sont issus d'algorithmes qui correspondent à des idées musicales, sont placés en haut du schéma : ce sont ceux qui portent le plus de significations. Nettement en dessous, toujours du côté synthèse, se trouvent le Sampler lorsqu'il est utilisé avec *addsamp*, puis les nombreux contrôles du Paf, et enfin *addsyn* pour le Module Additive¹⁶². Du côté des traitements les contrôles les plus significatifs sont ceux qui permettent la transposition (*hfre*), éventuellement retardée (*hdel*), puis ceux qui permettent la prolongation infinie (*revfb = 127* et *rgate*). Ensuite les contrôles qui permettent de décaler un spectre en fréquence (*fsfre*), et dernièrement les contrôles qui appartiennent plus à des Modules de mixage, ainsi de la Reverb quand elle n'est pas infinie (*revfb*), du Noise (*fnois*), et enfin de l'Harmonizer en mode flouteur.

¹⁶¹ Les modes secondaires des Modules de traitement sont précisés entre parenthèses.

¹⁶² Entre ces trois Modules on remarque une décroissance de « niveaux opératoires » (H. Vaggione) : le Sampler concerne des sons ayant déjà des timbres ; le Paf concerne des formants harmoniques ; et Additive concerne des sinusoides individuelles. L'association entre des niveaux opératoires sonores et des poids de signification n'est cependant pas systématique. Un module de synthèse additive dans une autre composition peut très bien avoir un haut poids de signification, bien qu'il soit d'un très faible niveau opératoire sonore. Il faut en effet différencier le sens des Modules dans la composition des niveaux opératoires.

Chapitre VIII

Outils informatiques pour l'analyse

Résumé

Ce chapitre présente quatre outils informatiques réalisés pour aider l'analyse musicale. Les trois premiers scannent les fichiers qlist afin d'en extraire différentes significations musicales, représentées sous forme textuelle. Le quatrième est plus spécifique, il s'attaque au problème du routing posé par la pièce. En appliquant différentes règles ce programme permet de savoir quels sont les Modules actifs à un événement donné et quelles sont leurs connexions : il donne une représentation du routing interne.

Préambule

Les programmes ont été réalisés avec le langage Processing¹⁶³. Néanmoins la présentation que nous faisons ici se fait sans insérer de lignes de code. En effet nous avons écrit ce chapitre de sorte que la compréhension de ces outils puisse se faire en dehors d'un langage de programmation donné. Le chapitre reste néanmoins très technique. Nous sommes en effet confrontés aux nombreuses écritures des qlist. Pour marquer une différence avec un niveau conceptuel, nous n'emploierons pas dans ce chapitre le terme *contrôle* – qui est attaché à la composition – mais celui de *variable* – qui lui ne précise rien quant à la composition.

Pour mener à bien la description des programmes il nous est nécessaire de communiquer au lecteur des fichiers de code et de données. Nous avons choisi de placer ces fichiers sur le web et d'indiquer leur chemin en note de bas de page lors des premières occurrences. De plus, pour tenter de faire face aux épreuves du temps, ces fichiers ont été regroupés et archivés sur le web¹⁶⁴.

¹⁶³ Processing est un langage de haut niveau dédié à la création multimédia, construit initialement sur le langage Java dans les années 2000 par deux chercheurs du MIT Media lab, Casey Reas et Benjamin Fry. Processing ne peut cependant se limiter à être *un* langage de programmation. En effet, comme il a été porté dans d'autres langages réputés (*e.g.* Javascript ou encore Python), il est plus approprié de l'envisager comme un moyen de concevoir la programmation dans la création multimédia – essentiellement les arts visuels.

¹⁶⁴ DOI : [10.5281/zenodo.1183938](https://doi.org/10.5281/zenodo.1183938)

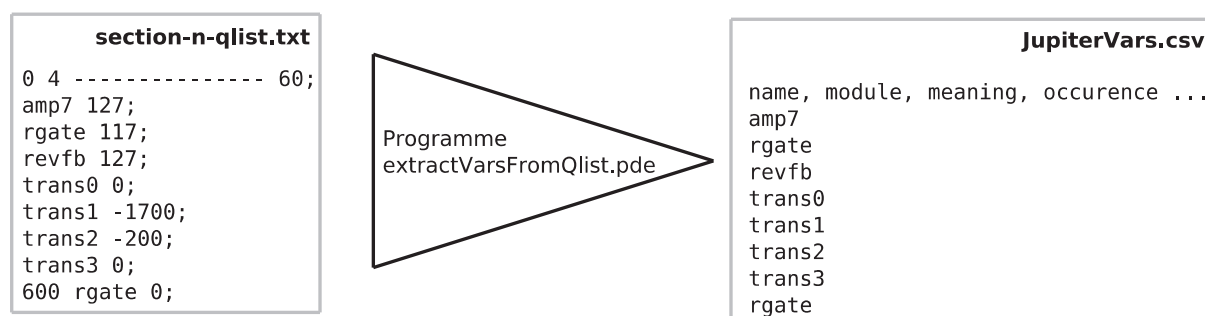
1 Extraire les données des qlist

1.1 Un tableau pour présenter les variables

Le premier outil réalisé permet d'identifier l'ensemble des variables présentes dans les qlist. Le programme fournit en sortie un tableau avec toutes les variables et quelques autres données qui viennent les décrire. Ce tableau vient faire face à la densité de l'artefact. Il nous a été des plus utiles dans la phase de découverte de l'artefact, qui consiste à se perdre et re-perdre dans les sous-patches et les abstractions, et relier mentalement les nombreux envois de données et audio ([send] et [send~]).

1.1.1 Aperçu

Le programme lit les 13 fichiers qlist au format texte et extrait un tableau général contenant toutes les variables identifiées, chacune accompagnée de descriptions.



FigVIII. 42 - Entrée et Sortie du programme extractVarsFromQlist

Les variables sont accompagnées de deux champs remplis manuellement – leur module d'appartenance (*module*), et leur signification (*meaning*) repris du chapitre précédent – et six remplis automatiquement : le premier est le nombre d'occurrences trouvées (*occurrence*) ; le deuxième les sections dans lesquelles la variable a été trouvée (*section*) ; le troisième le type de donnée de la variable (*dataType*) puis, si le *dataType* est un nombre alors les quatrième et cinquième indiquent les extremums (*min* et *max*), et si le *dataType* est une liste les septième et huitième donnent le nombre d'arguments (*argNb*) et la dernière valeur trouvée (*text*).

1.1.2 Résultat

Le programme a permis d'identifier pas moins de 243 variables. Elles sont présentes dans le fichier de sortie, *jupiterVars.csv*¹⁶⁵. En guise d'exemple nous présentons les 10 premières variables les plus utilisées (champs occurrence).

name	occurrence	section	dataType	min	max	argNb	text
chapo	1092	6,12	number	0	27		
addsyn	381	3,4	list			11	2 1 100 50 ...
rgate	270	all	number	0	137		
revfb	237	all	number	0	127		
fsfre	174	1,2,3,4,5,6,9, 11,12,13	number	-1000	109		
hamp1	164	all	number	0	127		
hamp2	164	all	number	0	127		
hamp3	158	all	number	0	127		
hamp4	154	all	number	0	127		
fneg	141	1,2,3,4,5,6,9,11, 12,13	number	0	127		

Tab. 16 - Extrait tronqué du tableau *jupiterVars.csv*, trié par occurrence¹⁶⁶

La variable *chapo* est celle qui a le plus d'occurrences, et de très loin, alors qu'elle est utilisée localement aux sections VI et XII. Cela s'explique par le fait que son niveau opératoire est bas – elle donne une valeur en fréquence – et qu'elle doit être appelée pour les 28 oscillateurs du Module (cf. Chap. VII 1.3.1). De la même manière on trouve en troisième position la variable *addsyn* qui donne aussi des valeurs en fréquence pour 20 oscillateurs (cf. Chap. VII 1.2).

La version intégrale du tableau (en ligne) montre aussi le cas particulier du Paf¹⁶⁷ qui regroupe pas moins de 90 variables : on ne pourra lire son écriture dans la continuité des lignes des qlist.

Enfin précisons que ce tableau nous a été d'une aide très précieuse pour le travail de pré-analyse, qui doit passer par l'intégration de ces nombreuses significations. Ce serait aussi une

¹⁶⁵ cf. [Jupiter/jupiterVars/jupiterVars.csv](#) (l'hyperlien utilise un visualiseur en ligne, s'il devient obsolète aller directement à l'adresse du tableau incluse dans le lien).

¹⁶⁶ Les colonnes *modules* et *meaning* ont été retirées pour optimiser l'affichage ; idem, le champ *text* pour la variable *addsyn* a été tronqué.

¹⁶⁷ Comprendre « du Module Paf ».

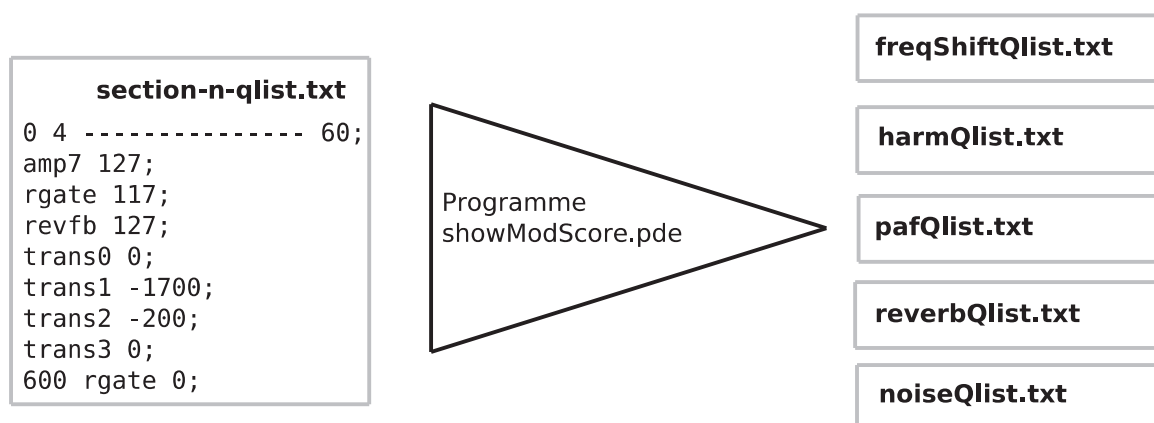
erreur de croire que ce programme, et le tableau qui en découle, ont eu très tôt l'apparence qu'ils ont dans cet écrit : l'accomplissement du programme et du tableau s'est fait au fur et à mesure de l'intégration de l'artefact, comme des reflets de notre appropriation.

1.2 Les partitions des Modules

Le deuxième outil informatique réalisé ne concerne plus l'artefact dans sa globalité mais les écritures d'un Module sur la durée de la pièce. Le programme extrait pour un Module toutes ses écritures. Le résultat obtenu doit nous permettre ainsi de répondre facilement à la question : « que fait tel Module à tel événement ? ». En l'état actuel les fichiers qlist ne nous permettent pas de répondre à cette question car (i) ils sont divisés pour chaque section, (ii) ils intègrent toutes les écritures (des Modules, du routing et de la configuration), et (iii) leur syntaxe est plutôt lourde.

1.2.1 Aperçu

Le programme réalisé permet d'extraire les données pour les Modules qui sont récurrents sur la longueur de la pièce. Il s'agit de la Reverb, de l'Harmonizer, du Frequency Shifter, du Paf et du Noise. Les variables recherchées sont celles que nous avons avancées au précédent chapitre, plus celles du routing externe (e.g. *nto2*).



FigVII. 43 - Entrée et sorties du programme showModScore

En entrée le programme reçoit les 13 fichiers qlist et en sortie il produit un fichier texte par Module¹⁶⁸. En plus de ces derniers, des fichiers sous forme de tableau, plus structurés, sont produits pour chacun des Modules.

¹⁶⁸ Cf. [Jupiter/showModScore](#)

1.2.2 Résultats

Les fichiers textes sont structurés sobrement : les données d'un événement sont indiquées après un alinéa ; les temps de délai sur une nouvelle ligne ; et tous les signes de ponctuation ont été retirés.

```

evt 7
  125
  trans1 0
  125
  trans2 -600
  trans3 -900
  trans4 -1900
  550
  hamp1 0
  hamp2 0
  hamp3 0
  hamp4 0

```

Les fichiers tableau sont, eux, plus classiques, avec les événements indiqués sur chaque ligne sous la forme *section.événement*.

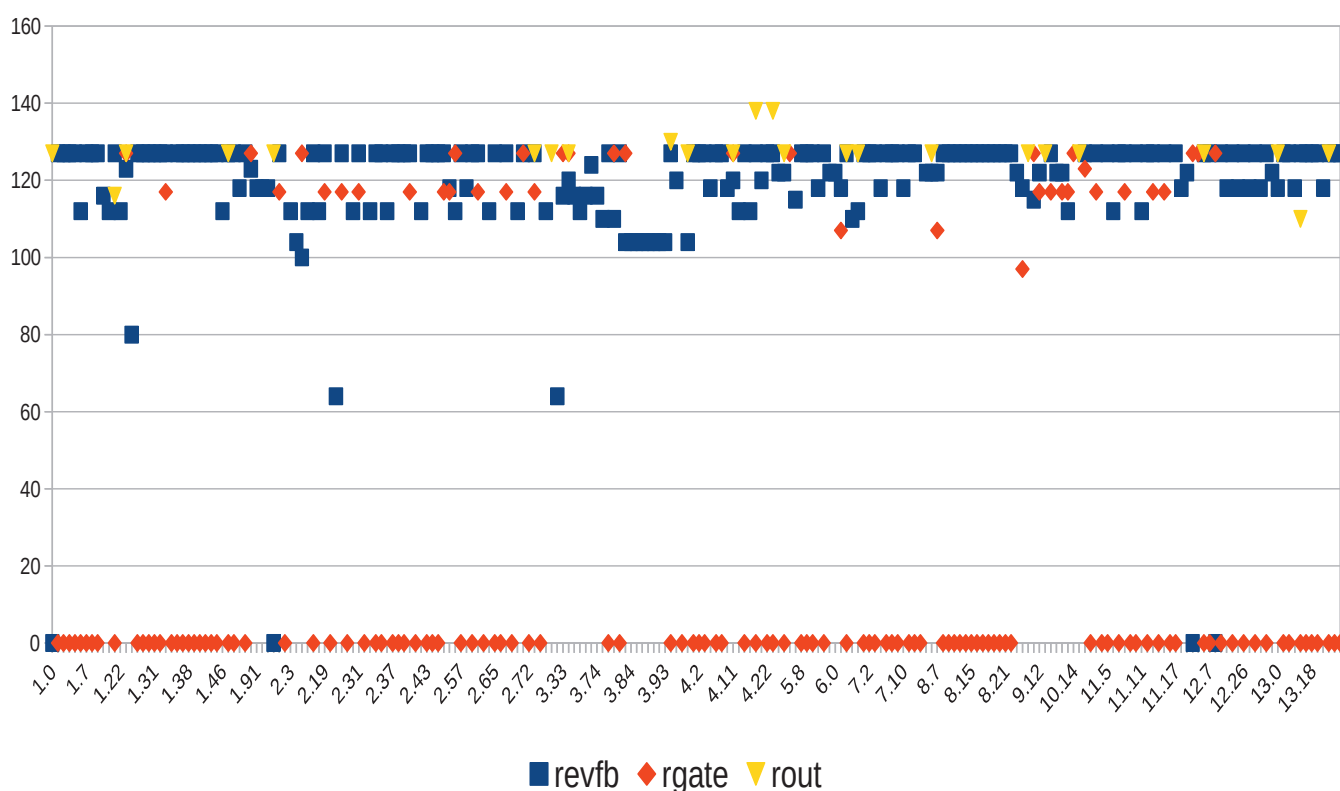
Section.evt	fneg	fpos	fsfre	fto2
3.89	127	127	10	
3.90	0	0		
3.92	0	0	3	100
4.0				113
4.1	127		494	
4.5		127	1109	102
4.8	127		127	
4.17	127	127	8	111
4.18			10	0

Tab. 17 - Extrait d'écritures structurées du FreqShift

Ces deux fichiers (textuel et tableau) nous permettent d'avoir un aperçu global de l'usage qui a été fait des Modules. Nous avons ainsi appris qu'une écriture n'est pas forcément nouvelle, mais qu'elle peut répéter la valeur précédente. Et dans la même lignée nous avons relevé des pratiques de « copier-coller ».

Ce sont les fichiers textuels qui nous ont été le plus utile pour le travail d'analyse musicale. Simplement, ils nous ont permis, avec la partition traditionnelle, de comprendre ce qui a été composé sur telle phrase ou sous-section.

Les tableaux nous ont été utiles pour les Modules avec peu de variables, comme la Reverb et le Noise. Un simple graphique sur l'évolution de leurs variables permet d'approcher globalement l'usage effectué par le compositeur.



*Fig. 44 - Écritures de la Reverb sur la durée de la pièce.
La durée est en abscisse sous la forme section.événement*

Le graphique ci-dessus montre par exemple que la Reverb possède une écriture très dense, sur toute la durée de la pièce. Cette densité s'explique avec un usage spécifique, où un son – une note de flûte souvent – va être prolongé indéfiniment ($revfb = 127$).

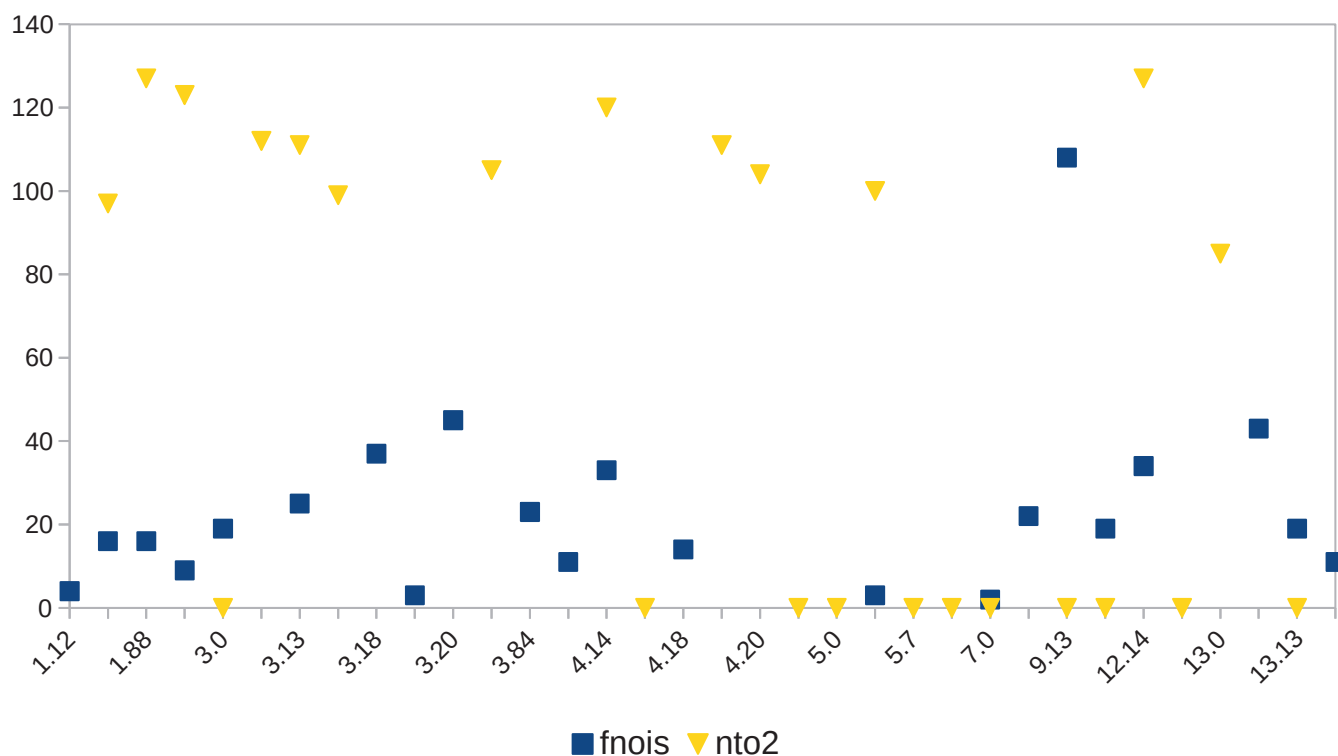


Fig. 45 - Représentation des écritures du Noise sur la durée de la pièce.
La durée est en abscisse sous la forme section.événement

À l'inverse le Noise possède une écriture dilatée. On relève aussi un usage quasi constant, où *fnois* reste en deçà de 60, ce qui aura pour effet de venir brouiller, perturber, le signal sur lequel il s'applique¹⁶⁹.

Plus substantiellement ce graphique amène la question du *routing interne*. Une lecture trop rapide relèverait que le Noise est éteint du début de la section V jusqu'à la XII, et de plus que *fnois* reçoit, durant cette même période, sa plus forte valeur (~120). Or, si pour les Modules de synthèse une lecture des fichiers obtenus peut nous indiquer si le Module est actif ou non, s'il produit du signal, il n'en va pas de même pour les Modules de traitement. Dans notre exemple le Noise peut être actif tout le long de la pièce, nonobstant *nto2*, avec par exemple une sortie dans le Spat. En conséquence pour induire des connaissances sur l'état des Modules il est nécessaire de s'attaquer, inévitablement, au *routing interne*. Avant de présenter le travail réalisé sur le routing nous exposons ci-après un autre programme qui vient répondre au problème du Paf.

¹⁶⁹ cf. Chap VII 2.4

2 Représenter les données du Paf

Comme nous l'avons vu au précédent chapitre le Module Paf est très particulier. Ceci vient du fait qu'il est imposant (8 voies, synthèse PAF, frequency shifting, vibrato et arpeggiator) et d'autre part qu'il ne possède pas de méthode raisonnée pour prescrire des valeurs à la cinquantaine de variables qu'il implique¹⁷⁰. Le programme réalisé a pour but de faire face à cette quantité de variables. Il doit permettre à l'analyste de connaître l'état du Paf sans passer par la lecture de toutes ses variables éparpillées.

Le point central de ce travail consiste en la réalisation d'une *représentation* du Module Paf, ou plus justement une représentation de chacune de ses voies. Le terme représentation pourrait paraître impropre dans la mesure où notre résultat n'a rien de graphique. Nous souhaitons néanmoins l'utiliser pour deux raisons. D'abord parce que c'est le résultat du programme même qui est ensuite utilisé de nouveau pour savoir s'il y a eu un changement d'état : notre résultat rend présent le Module dans le programme. Ensuite parce que le résultat obtenu est d'un plus haut niveau d'abstraction que le précédent fichier *pafQlist.txt*. Par exemple, en plus de concaténer des données, les valeurs MIDI sont données en noms de notes.

Les représentations que nous avons obtenues permettent ainsi d'inférer des connaissances sur le Module bien plus rapidement que ne le permet le fichier *pafQlist.txt*.

2.1 Aperçu

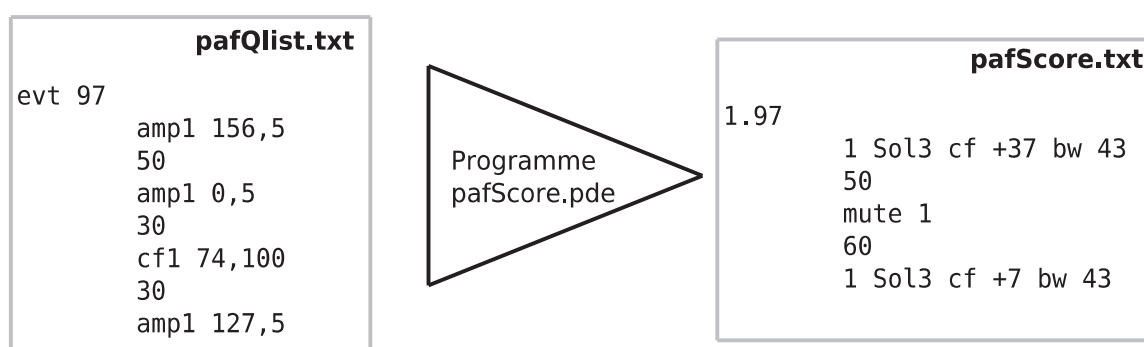


Fig. 46 - Entrée et sortie du programme pafScore

Le programme reçoit en entrée le précédent fichier *pafQlist.txt* et extrait un nouveau fichier texte, *pafScore.txt*¹⁷¹. Quantitativement, nos représentations permettent de réduire l'information en passant d'un fichier de 1900 lignes à un de 900.

¹⁷⁰ La Paf a 8 canaux nécessitant 7 variables. Les 90 variables identifiées précédemment incluaient celles du routing, qui ne sont pas visées ici.

¹⁷¹ cf. [Jupiter/showModScore/pafScore/pafScore.txt](#)

2.2 Fonctionnement

Les représentations ont été obtenues en utilisant la Programmation Orientée Objet (POO par la suite). On peut comprendre cette méthode comme un moyen pour affronter la structuration des Modules. Par exemple l'Harmonizer se compose de 4 voies dont chacune nécessite 4 variables. Dans le cas du Module Paf le problème est plus flagrant : *le Module Paf est composé de 8 voies dont chacune nécessite 7 variables*¹⁷². Nous avons alors reproduit la même structuration en POO. La classe Paf appelle la classe Channel, qui elle-même appelle la classe Var¹⁷³. Ainsi, lorsque la classe Paf est instanciée en début de programme 8 autres instances Channel sont créées, qui elles-mêmes créent 7 autres instances de la Classe Var. Le schéma UML¹⁷⁴ ci-dessous rend compte de cette structuration.

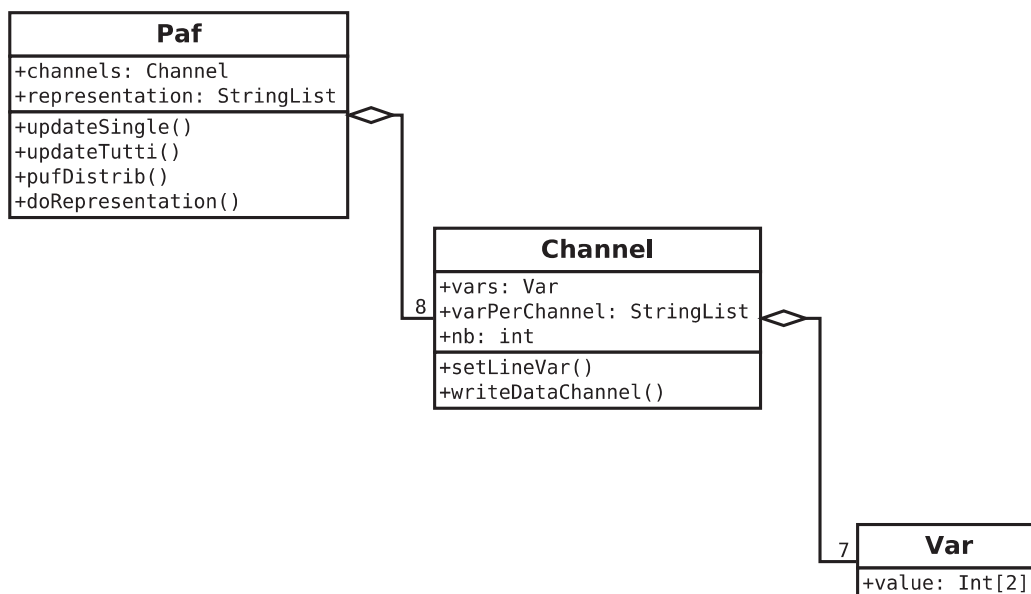


Fig. 47 - Schéma UML, structuration du programme pafScore.pde

En plus des relations entre classes, le schéma présente, dans les deux encadrés du dessous, leurs *attributs* et leurs *méthodes*¹⁷⁵.

¹⁷² Parmi les 9 variables d'une voie du Paf (cf. Chap. VII 1.4.4), nous ignorons ici *noise* (qui relève du routing) et *cf-line* (un temps de lissage).

¹⁷³ En POO le nom d'une classe commence par une majuscule.

¹⁷⁴ Universal Modelling Language

¹⁷⁵ On note *a posteriori* que la classe Var pourrait être remplacée par une classe nativement présente dans Processing, StringDict. En effet Var était utile pour prendre en compte les éventuels temps de lissage des valeurs, mais cette idée fut abandonnée puisque de lissage n'est pas utilisé pour le Paf.

2.2.1 Repérer les changements d'état

Le programme intègre une sorte d'intelligence puisque avant de donner une représentation il vérifie s'il y a eu un changement ou non avec l'état précédent. Si oui, la représentation est donnée sinon, le programme passe à l'événement suivant. Le fichier de sortie contient ainsi uniquement les changements d'états des voies du Paf.

2.3 Difficultés

Deux difficultés se présentent. La première vient de la comparaison entre différents états du Module. Avec cette comparaison il n'est plus question d'écrire les données à chaque événement, ou chaque délai, mais bien de savoir s'il y a eu ou non un changement avec l'état précédent. Un système avec un tampon (*buffer*) doit alors être intégré. L'autre difficulté vient des différents moyens utilisés pour prescrire des valeurs au Paf. Il y a des variables seules (e.g. *amp3*), puis ce que nous appelons des *variables de substitution*. Par exemple la variable *amptutti4* se substitue aux variables *amp1 ... amp4*, ou encore *pufamp* qui se substitue à *amp1 ... amp8*¹⁷⁶.

2.4 Résultat

Nous avons opté pour la représentation suivante :

```
3.40
mute 5 6 7
1 Mi3 cf +24 bw 54
2 La3 cf +18 bw 59
```

Le numéro précédant la note de musique est le numéro de la voie. L'essentiel de cette représentation tient ensuite sur trois points :

- l'*amplitude* est vue uniquement comme un interrupteur : soit le canal est actif soit il ne l'est pas. Puis, lorsque les canaux s'éteignent leur numéro est indiqué après *mute*.
- Le *pitch* est donné en nom de note – ce qui facilite la lecture avec la partition gravée.
- Le *cf* est donné par positionnement selon le *pitch*.

À l'exception du *pitch*, toutes les autres données sont inscrites en note MIDI, comme dans les qlist.

¹⁷⁶ Ces trois moyens de prescrire des valeurs au Paf correspondent aux trois premières méthodes de la classe Paf : *UpdateSingle()* pour des variables seules ; *updateTutti()* pour des groupements ; *pufDistrib()* pour l'arpeggiator.

3 Le routing

3.1 Introduction

Le dernier programme réalisé concerne exclusivement le routing. Comme nous l'avons vu le routing de *Jupiter* est particulier puisqu'il est *persistant* : il se fait dans la succession des événements. Ainsi, pour connaître une configuration du routing, l'analyste doit remonter les données des événements précédents. Le programme réalisé vient faire face à ce problème. Il doit permettre de répondre à la question « quels sont les Modules actifs à tels événements et avec quelles connexions ? ».

Répondre à cette question nous a demandé un travail conséquent réparti en trois étapes, trois programmes. La première étape trouve les « pré-états » de tous les Modules ; la deuxième relève à partir des qlist toutes les connexions entre Modules dans la continuité de la pièce ; la dernière applique un ensemble de règles afin de trouver les différentes configurations du routing.

Avant de commencer précisons que nous avons travaillé avec une précision de l'ordre des événements, c'est-à-dire que les temps de lissage ont été ignorés.

Enfin, l'essence du routing tient dans le typage des Modules, certains sont de synthèse et d'autres de traitement. Pour alléger nous les désignons avec les expressions « les *synth* » et « les *treats* ».

3.2 (i) Trouver les (pré-)états des Modules

Cette première étape permet de savoir si les Modules sont actifs ou non. Mais il ne peut s'agir ici que d'un état premier (pré-état) puisque, comme nous l'avons vu avec le Noise, avant de pouvoir affirmer qu'un Module est actif il faut vérifier son acheminement à la diffusion. Cette étape a ainsi lieu à l'échelle des Modules et non de leurs connexions. Pour les *synth* il s'agit de savoir si du signal est produit ; pour les *treat* de savoir s'ils sont aptes à en produire.

Pour les *synth*, à l'exception du Paf, nous avons dû faire une détection manuelle essentiellement à partir des qlist, et parfois avec la partition et l'écoute de la pièce. Pour les *treat* nous avons réalisé un programme qui effectue une détection automatique.

3.2.1 Détection manuelle

3.2.1.a Sampler

La détection du Sampler s'est faite selon les deux utilisations que nous avons identifiées : soit à partir de la variable *addsamp* soit à partir de lecture de fichiers (e.g. *ost6-start*).

Pour *addsamp* les états ont été déduits manuellement à partir des qlist. La déduction s'est aussi faite avec des allers-retours vers le patch puisqu'il y a des variables *ad hoc* qui déclenchent des métronomes affectant *addsamp*¹⁷⁷. Afin de connaître le pré-état des Modules la simple règle suivante à été appliquée : si des écritures sont présentes dans l'événement alors le Module est *on* et inversement. En effet le Sampler a ceci de particulier que le signal qu'il produit est nécessairement limité dans le temps. Même si *addsamp* concatène la variable *duration*, cette durée ne peut en effet dépasser celle des samples, qui sont de 700 ms non transposés¹⁷⁸.

Pour l'utilisation à partir de fichiers nous avons utilisé l'écoute et la partition traditionnelle. La détection est facile puisque lorsqu'il produit du signal, le Sampler est le seul *synth* actif, et surtout parce que son rendu est très marqué.

3.2.1.b Additive et Chapo

Pour ces deux Modules la détection est similaire à celle effectuée avec *addsamp* : nous avons lu les occurrences de *addsyn* et *chapo*. La différence essentielle réside dans le fait que les voies de ces Modules sont actives jusqu'à ce qu'une écriture vienne les éteindre. Il faut donc bien s'assurer que toutes les voies d'un Module sont éteintes avant d'indiquer que le Module est *off*.

Le Module Additive, comme nous l'avons vu, a une utilisation très ciblée à l'intérieur de la section III, et superposée à *addsamp*. Le relevé a ainsi été simple : lorsque les occurrences correspondent à l'extinction des 12 voies du Module, alors il est *off*.

La détection du Chapo a aussi été aisée puisqu'il est utilisé continuellement sur les sections VI et XII, très singulières. Le Module est *off* lorsque ses 20 voies sont éteintes.

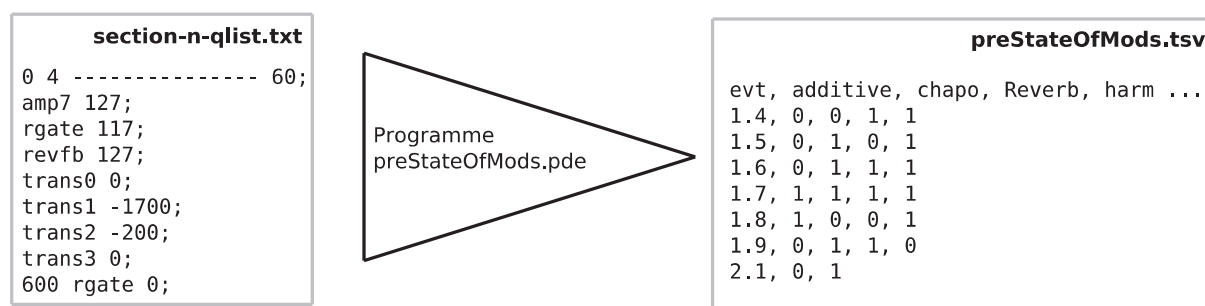
3.2.2 Détection automatique

Pour les *treat* et le Paf nous avons réalisé un programme qui déduit automatiquement les états de ces derniers après avoir scanné les qlist. La déduction des états est faite pour tous les événements qu'il y ait ou non des écritures des Modules.

¹⁷⁷ cf. les variables *evt68*, *evt76*, *evt79*, *evt80* etc., et *sec2-evt77* et *sec2-evt78* dans le fichier *JupiterVars.csv*.

¹⁷⁸ Les samples sont définis sur 30 k échantillons.

3.2.2.a Aperçu

Fig. 48 - Entrée et sortie du programme *preStateOfMods*

Le programme reçoit en entrée tous les fichiers qlist puis, après l'application de règles, le fichier *preStateOfMods.txt*¹⁷⁹ est construit. C'est un tableau où les événements s'enchaînent sur les lignes et les Modules sur les colonnes. À l'intersection de ces deux dimensions l'état du Module est donné par une variable booléenne.

Le programme réalisé est semblable au précédent réalisé pour le Paf. Il se divise en deux parties. Une première, « chariot », où les qlist sont scannés ligne par ligne ; une seconde qui permet de structurer les données recueillies.

3.2.2.b Fonctionnement

Tous les Modules que nous visons possèdent une certaine structure de données commune (*e.g.* nom, type de Module, variables, canaux, variables de sortie). En utilisant la POO nous avons alors créé une classe, nommée *Module*, qui fixe cette structure de données. Celle-ci est attachée à une seconde, nommée *DictList*¹⁸⁰, qui lie une variable à une liste de nombres – le mot *fpos* lié à toutes ses occurrences par exemple.

¹⁷⁹ [lupiter/routing/a_preStatesOfMods/preStatesOfMods.tsv](#)

¹⁸⁰ Le nom *DictList* prend sens avec le référentiel de Processing. Comprendre « Dictionary of list ». Une classe qui relie un type de donnée *String* à un type de donnée *IntList*, une suite de nombres entiers.

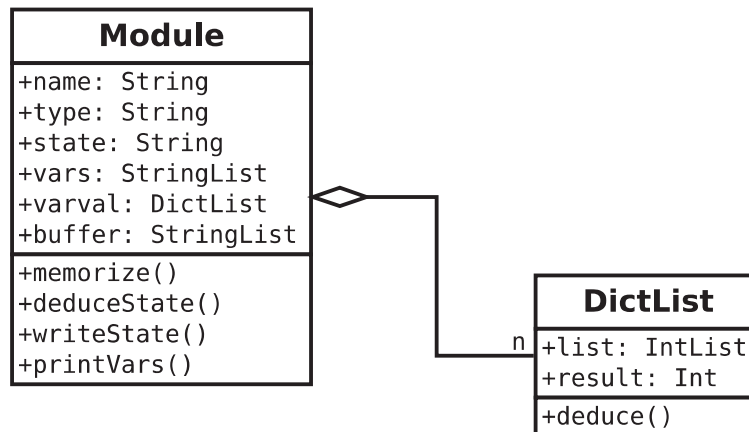


Fig. 49 - Schéma UML de la structuration du programme *PreStateOfMods.pde*

Le programme s'initie en créant des instances de la classe Module pour les 4 Modules à analyser. Le n du schéma indique que le nombre de variables attachées aux Modules n'est pas fixe. En effet ce nombre varie selon le Module, comme en témoigne la table ci-dessous.

Module	Type	Variables
paf	synth	amp1, amp2, amp3, amp4, amp5, amp6, amp7, amp8
reverb	treat	rout
harm	treat	hamp1, hamp2, hamp3, hamp4
freqShift	treat	fpos, fneg

Tab.18 - Données des Modules utilisées pour les instances

Le programme peut alors se comprendre de façon approximative : toutes les lignes sont scannées ; lorsqu'une des variables du précédent tableau est identifiée, sa valeur est enregistrée dans une liste de nombres propre au Module ; lorsqu'un événement est fini, le contenu de ces listes est analysé de sorte à déduire l'état du Module.

Règles

La déduction de l'état s'est faite avec trois règles : si durant un événement la variable de sortie d'un Module¹⁸¹

- a eu au moins une occurrence supérieure à 0, alors le Module est actif.
- n'a eu que des occurrences égales à 0, alors le Module est off.

¹⁸¹ Afin de simplifier nous considérerons ici que les Modules n'ont qu'une seule variable.

- n'a eu aucune occurrence, alors le Module reste dans l'état précédent.

3.2.2.c Difficultés

Les difficultés viennent des variables de substitution qu'il faut traiter au cas par cas (*viz.* $amptutt\{4,5,6,8\}$, $env[1-8]$, et $trans[1-4]$).

3.3 (ii) Extraire le Routing

La deuxième étape concerne le routing interne et externe. Son but est d'extraire des qlist les connexions effectives des Modules, aussi bien entre Modules (routing interne) que des Modules aux haut-parleurs (routing externe). Si la précédente étape permettait de répondre à la question « quels Modules sont aptes à produire du signal ? », la seconde doit permettre de répondre à « quelles sont les sorties effectives des Modules ? ». Comme le programme réalisé est très similaire au précédent nous effectuons une présentation succincte.

La même structuration est en effet reprise. Les Modules et les variables de l'artefact instancient, réciproquement, les classes Module et DictList. La table ci-dessous présente les instances créées.

Module	Type	Variables
flute	synth	dtor, dtoh, dtof, dton, dto4
sampler	synth	ttor, ttoh, ttof, tton, tto4, tto2
osc	synth	otor, otoh, ofof, oton, oto4, oto2, noise1, noise2, noise3, noise4, noise5
reverb	treat	rtoh, rtof, rton, rto4, rto2
harm	treat	htor, htof, hton, hto4, hto2
freqShift	treat	ftor, ftoh, fton, fto4, fto2
noise	treat	ntor, ntoh, ntof, nto4, nto2

Tab. 19 - Données Modules utilisées pour les instances

Le fonctionnement du programme tient essentiellement dans la compréhension du tableau produit, *modsConnection.txt*¹⁸². Les événements s'enchaînent sur les lignes et les 7 Modules à scanner sur des colonnes. L'indication d'une connexion se fait de la façon suivante : si le programme analyse une connexion effective (e.g. $rtoh = 120$) à l'événement 2.40 alors, à la ligne correspondante et à la colonne nommée *reverb*, il inscrit « rtoh », et ainsi de suite.

¹⁸² [Jupiter/routing/b_modsConnections/modsConnection.tsv](#)

Les difficultés de réalisation viennent du regroupement des Modules Paf, Additive et Chapo dans ce faux module « osc », et, corrélativement, des connexions du Paf vers le Noise qui se font à l'échelle des canaux (*viz. noise*{1,2,3,4,5})¹⁸³. Comme précédemment, il a été nécessaire de faire du cas par cas pour des variables de substitution, dont le nom commence ici par *sto*. Ces dernières distribuent leurs valeurs aux Modules de synthèse, c'est-à-dire au Sampler et au « osc » : *sto2* affecte *tto2* et *oto2*, *stof* affecte *ttof* et *otof*...

3.4 (iii) Déduire les états des Modules et le routing

Cette ultime étape doit permettre de répondre aux questions initiales : à un événement donné quels sont les Modules actifs et quels sont leurs connexions ? C'est à partir des réponses à ces questions que l'analyse pourra en effet être réalisée. Ce programme se base sur les résultats des deux précédentes étapes. Par ailleurs, c'est aussi le plus dense puisqu'il nous a été nécessaire de concevoir un algorithme qui investit les nombreuses connexions du routing.

3.4.1 Aperçu

Le programme reçoit les deux précédents fichiers, *preStateOfMods.tsv* et *modsConnection.tsv* et, après application d'un certain nombre de procédures, produit les fichiers *statesOfMods.tsv*¹⁸⁴ et *routingChains.txt*¹⁸⁵. Le premier indique dans un tableau l'état de tous les Modules, le second indique textuellement l'état du routing. C'est ce dernier fichier qui nous a été le plus utile pour l'analyse.

3.4.2 Fonctionnement

Une seule classe est utilisée. Elle définit la structure de données commune à tous les Modules.

Les attributs *inputs* et *outputs* permettent de stocker les différentes connexions des Modules ; les attributs booléens *dac* et *state* de stocker la connexion au routing externe et l'état du Module.

Le programme s'initie en créant des instances pour chacun des Modules (Flute, Sampler, Osc, Reverb, Harm, FreqShift, Noise, Spat) puis, à partir de la table *preStatesOfMods.tsv*, il sélectionne les Modules actifs événement par événement.

L'affectation des connexions entre Modules se fait ensuite avec la table *modsConnection.tsv*. Par exemple, la lecture à un événement donné de *rtoh* ajoute le nom «harm» à l'attribut *outputs* de l'instance *reverb* et, réciproquement, le nom « reverb » est ajouté à l'attribut *inputs*

183 cf. Chap. VII 1.4.5

184 [Jupiter/routing/c_statesAndRouting/statesOfMods.tsv](#)

185 [Jupiter/routing/c_statesAndRouting/routingChains.txt](#)

de l'instance *harmonizer*. De cette façon il est possible d'avoir une représentation interne du routing dans son entièreté.

Module
<pre>+name: String +type: String +inputs: StringList +outputs: StringList +dac: Boolean +state: Boolean</pre>
<pre>+reset() +setInputOutput() +treatWithoutInputs() +deduceState() +removeDeadConnections()</pre>

Fig. 50 - la classe Module

Dans la réalisation du programme il nous a été nécessaire d'avoir un retour sur cet état du routing ; nous avons opté pour la syntaxe suivante :

```
flute out(reverb)
noise in(osc, reverb) out(spat)
spat in(noise, osc, reverb)
osc out(noise, spat)
reverb in(flute) out(noise, spat)
```

3.4.2.a Règles

Une fois les connexions établies pour un événement un certain nombre de règles sont appliquées dans l'optique d'obtenir un routing cohérent. En effet, comme Alain Bonardi l'avait déjà observé¹⁸⁶, il réside dans les écritures du routing des incohérences. Par exemple une connexion de la Flûte vers le Noise alors que ce dernier n'a aucune sortie – *i.e.* il est *off*. Deux règles – la seconde est présente dans la méthode *treatWithoutInputs()* – sont appliquées :

- si le Module n'a aucune sortie, *i.e.* ni dac ni routing interne, alors son état est *off*
- si le Module, hormis la Reverb¹⁸⁷, est un *treat* et n'a pas d'entrée, alors son état est *off*

¹⁸⁶ *e.g.* Alain Bonardi, «Analyser et représenter la production du son dans les œuvres mixtes. Exemple du patch Max/MSP de *En Echo* de Philippe Manoury », présentation orale dans le cadre du séminaire *Analyse par segmentation des musiques électroacoustiques*, Saint Étienne, 2013.

¹⁸⁷ Avec son mode infini la Reverb peut en effet être active même si elle n'a pas d'entrée.

Les états « finaux » des Modules sont ensuite déduits – la méthode *deduceState()* – et les connexions mortes qui pointent vers un Module *off*, sont retirées – la méthode *removeDeadConnections()*.

3.4.2.b Chaînes de routing et algorithme

La dernière représentation du routing que nous avons vue est nettement trop lourde pour une lecture humaine. Plutôt que de penser le routing à partir des Modules nous l'avons investi avec le principe de *chaîne*. Une chaîne commence par un *synth* et se finit soit dans un dac soit dans le Spat.

La transformation de la dernière représentation vers une autre construite avec le principe de *chaîne de routing* est effectuée par un algorithme. Ce dernier doit, tout en mémorisant leurs noms et leurs successions, traverser toutes les instances actives interconnectées. Il permet ainsi de produire la représentation textuelle suivante :

```
10.12
flute>spat
sampler>reverb
sampler>harm>noise>spat
sampler>harm>reverb
dac : harm, reverb, sampler
```

Enfin, comme pour le programme *patScore*, cette représentation du routing est aussi utilisée pour savoir s'il y a eu des changements avec l'événement précédent. Le fichier final *routingChains.txt* contient donc cette représentation en chaîne lorsqu'il y a eu des changements. Avec un saut de ligne pour chaque événement, le fichier produit contient pas moins de 1700 lignes.

3.4.3 Difficultés

La principale difficulté vient de l'algorithme réalisé. Celui-ci n'est pas évident puisqu'il doit en fait être récursif, c'est-à-dire que le chemin à parcourir d'instance en instance n'est pas donné à l'avance, mais qu'il se dessine au fur et à mesure de son parcours. Dans le précédent exemple, une fois la chaîne *sampler>reverb* trouvée, il faut remonter au Sampler pour investir l'autre connexion qui va à l'Harmonizer, et ainsi de suite. Lorsque l'algorithme se trouve à une instance donnée, toutes les connexions qui ne sont pas investies sont à placer dans un tampon, et parallèlement le chemin parcouru qui a mené à cette instance est aussi à mémoriser. Une description plus détaillée de l'algorithme est donnée dans l'annexe C.

3.5 Résultat

3.5.1 Les Modules actifs

Le tableau produit nous a été utile pour effectuer une représentation des Modules actifs, sur la durée de la pièce.

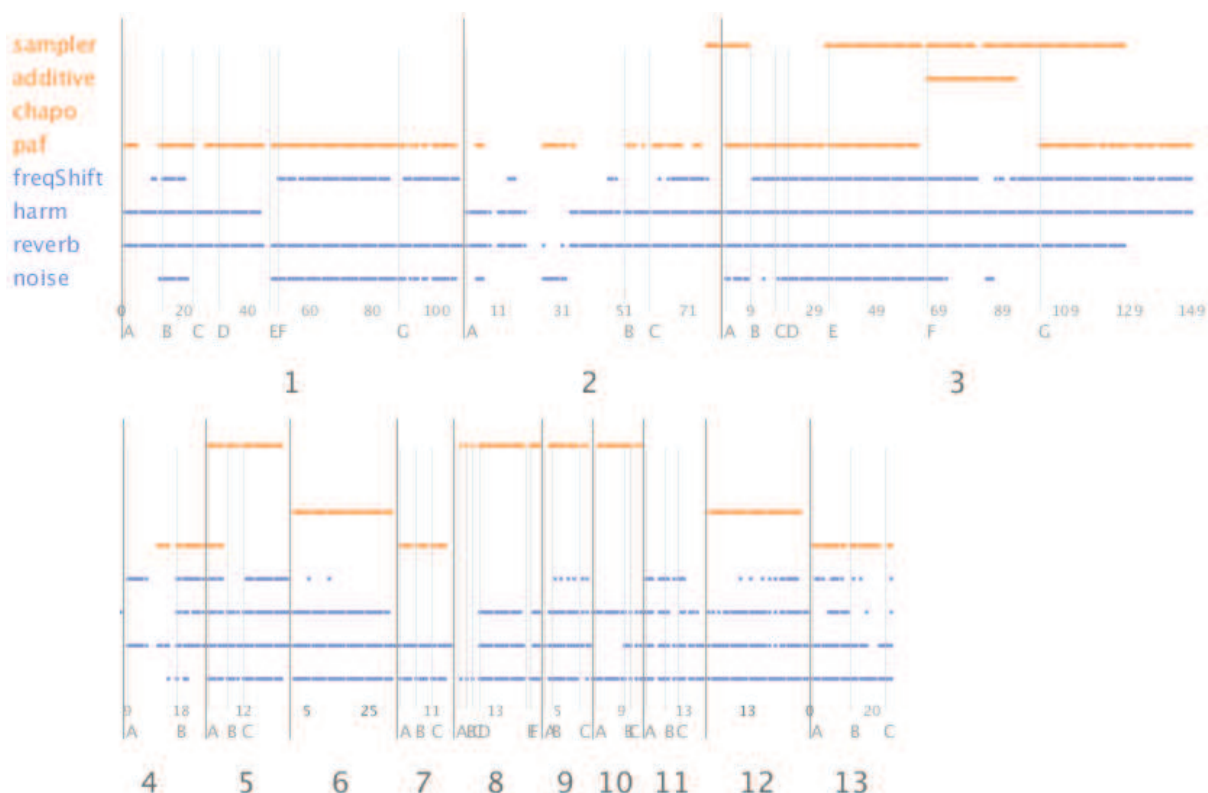


Fig. 51 - États des Modules sur la durée de la pièce.
Image découpée pour l'affichage.

Les 8 Modules sont présents en ordonnée avec la typologie *synth* et *trat* marquée en couleur. Le temps est présent en abscisse selon la succession des événements et non des minutes¹⁸⁸. La segmentation de la partition traditionnelle est reprise pour l'abscisse : du bas vers le haut on retrouve, les numéros des sections, les lettres des sous-sections, puis les numéros des événements.

Cette représentation permet d'avoir un aperçu sur l'état des Modules dans la pièce. Elle montre que les Modules de traitement sont les plus présents. Accompagnée d'une écoute, elle permet d'effectuer deux regroupements de sections. Celles où le Chapo est actif, la VI et XII, et

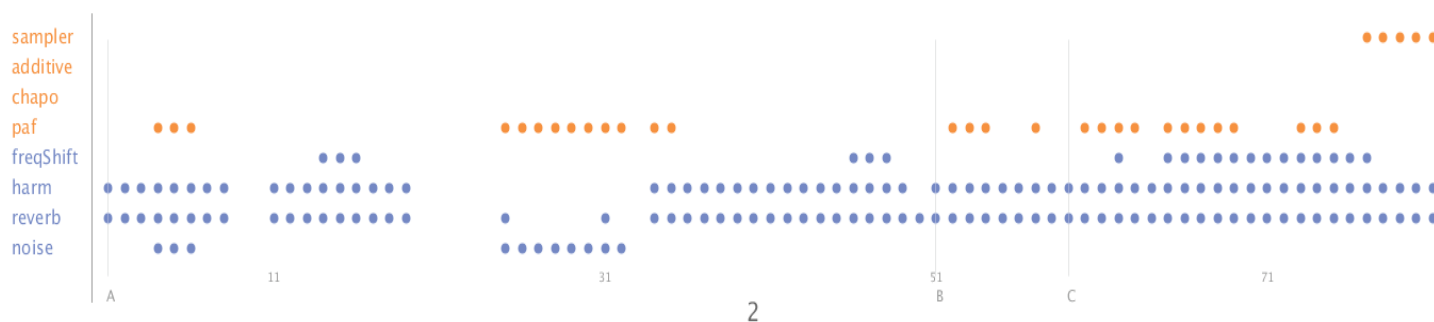
¹⁸⁸ Ainsi la section III occupe le plus de place tandis que c'est la VI qui est plus longue en durée (cf. Chap VII 3.2).

un groupe plus vaste, I, II, IV, VII et XIII, où l'on retrouve les mêmes Modules. On relève aussi la spécificité de la III où, à l'exception du Chapo, tous les Modules sont requis.

Il faut cependant rester très prudent avec une telle représentation. Il est en effet nécessaire de l'accompagner de l'écoute et/ou de la partition traditionnelle afin qu'elle puisse être pleinement signifiante. Ce qui lui manque essentiellement c'est l'usage des Modules qui a été fait. Par exemple le Sampler dans la III n'est pas le même que dans la V, qui est encore différent de celui de la VIII. La structuration de la pièce est abordée au prochain chapitre après notre l'analyse (cf. Chap IX 3).

3.5.2 Modules actifs par section

Le même type de représentation a été réalisé pour toutes les sections.



*Fig. 52 - États des Modules sur la section II
un point est un événement*

Ces représentations nous ont été utiles pendant le travail d'analyse, lors des premiers contacts avec les sections afin d'avoir un aperçu sur leur construction.

3.5.3 Les chaînes de routing

Le fichier textuel représentant les chaînes de routing a été le plus utilisé. Il permet au niveau local des événements de comprendre l'embranchement entre Modules – ce qui manquait dans les précédentes représentations.

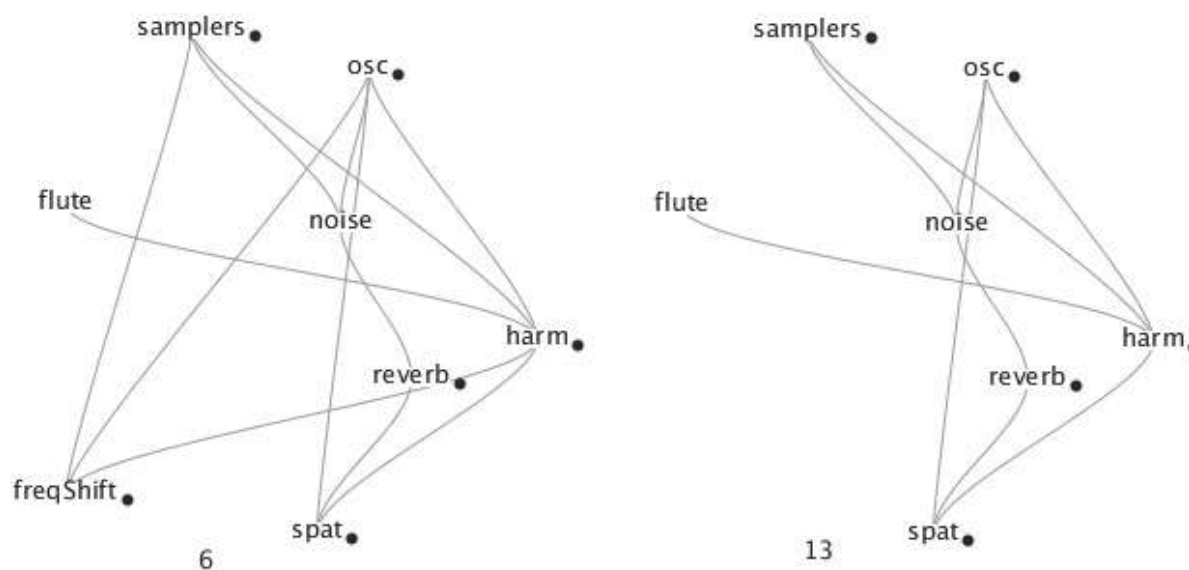


Fig. 53 - Représentation du routing. Section VI événement 6 et 13
Les modules suivis d'un point noir sont reliés à la diffusion (dac)

Nous avons aussi utilisé ce fichier textuel pour faire des représentations, générées automatiquement événement par événement. Ces représentations nous ont cependant été très peu utiles¹⁸⁹. Elles ont servi avant de commencer le travail d'analyse, afin de comprendre la conception du routing durant la composition.

Le fichier texte obtenu et ces dernières représentations nous ont montré en fait que la précision obtenue était trop grande comparée à l'usage du routing qui a été fait dans la composition. C'est typique par exemple pour le Noise qui, comme nous l'avons vu dans la structure de contrôle de l'artefact, n'est pas un Module « support de la composition », mais un module de mixage : il vient brouiller un signal qui est déjà diffusé. Cela se retrouve aussi dans les nombreuses connexions des Modules au dac. On s'aperçoit qu'il y a *un* usage récurrent des *treat* qui consiste moins à créer une nouvelle sonorité que de venir enrichir, brouiller, une sonorité déjà diffusée. Cet usage-là, n'étant pas porteur de grande signification, doit être dépassé par l'analyse musicale. Une analyse qui se baserait sur les interconnexions entre le Noise, la Reverb (en dehors du mode infini) et le FS resterait lacunaire. Dans le même temps, et c'est fondamental, cet usage-là participe à la sonorité de *Jupiter*. Les ignorer serait une erreur, comme en atteste leur forte présence sur la durée de la pièce. Cet usage des *treats* s'éclaire bien avec une métaphore que le compositeur nous a transmise¹⁹⁰ : « les *synth* sont des autoroutes, droites, avec un goudron bien lisse et un paysage constant ; les *treat* permettent de faire face à cette uniformité, de sorte que l'on ne s'ennuie pas dans la conduite, dans l'écoute. »

¹⁸⁹ Nous les avons générées uniquement pour la VI où le routing reste assez stable.

¹⁹⁰ Métaphore partagée lors d'un entretien [Manoury, 2016].

Le fichier texte nous a ainsi été utile pour relever le routing des sous-sections et sections avec une lecture experte, où il est possible de distinguer parmi tous ces changements ceux qui sont importants de ceux qui ne le sont pas. Ce fichier nous a aussi été très utile pour éclairer des sonorités surprenantes (*e.g.* Section VII événement 1 à 16), qui s'expliquent, en plus de la configuration des Modules, par celle du routing.

4 Report des résultats

Au cours de l'analyse nous avons reporté les résultats obtenus de ces précédents programmes sur la partition traditionnelle. Évidemment la confrontation de ces deux types de représentation – la première « textuelle » qui résulte des précédents programmes informatiques ; la seconde « graphique » qui est celle de l'écriture occidentale – pose d'importantes questions dans le champ théorique des représentations musicales. Ces questions ne peuvent cependant être traitées ici pour la simple raison qu'elles méritent un travail substantiel – que nous prévoyons d'effectuer dans les prochaines années. Nous précisons dans ce grand titre simplement le report que nous avons effectué, non systématiquement, des précédentes données textuelles sur la partition traditionnelle.

4.1 Les écritures des Modules

Dans la partition traditionnelle toutes les écritures relatives aux Modules sont présentes dans un système de deux portées nommé S.T.R., « Système Temps Réel ». Nous avons alors reporté les écritures des Modules résultant de nos travaux (*freqShiftQlist.txt*, *harmQlist.txt*, *pafScore.txt* etc.) sur ce même système à l'aide d'un simple code couleur.

The image shows a musical score for a Flute and an S.T.R. (Système Temps Réel) system. The S.T.R. system is labeled 'S.T.R.' and 'IA'. It features a sequence of notes with colored circles (1, 2, 3) and a bracket labeled '9'. The Flute part is in 3/4 time, marked 'mf', 'f', 'p', 'sffz', 'sffz', 'mf', and 'molto vibr.'. The S.T.R. system has a tempo marking '♩ = 72' and a note '(Harm. → Rev. ∞)'.

Fig. 54: Le système STR de la partition traditionnelle et le code couleur utilisé

Orange fait référence à l'Harmonizer ; Vert à la Reverb ; Bleu au Paf⁹¹ ; Violet au FS et enfin Marron au Sampler. De plus nous avons indiqué, le cas échéant, le mode d'utilisation des Modules : *flou* pour l'Harmonizer en mode floutage ; *hdel* toujours pour l'Harmonizer quand il ajoute des délais conséquents ; *RM* pour le FS en mode Ring Modulation. La Reverb en

191 Ou pour le Chapo ou pour Addsyn ; en sachant que ces Modules ne sont pas actifs en même temps.

mode infinie est elle précisée en colorisant en vert le cercle de l'événement ; un « c » vert ensuite indique la fermeture de la porte de la Reverb (*rgate*).

4.2 Le routing

Le routing est représenté sur la partition traditionnelle sous forme textuelle en une seule ligne. Quand cette indication correspondait à nos résultats (le fichier *routingChains.txt*) nous avons simplement augmenté cette indication du code couleur et des éventuels Modules non pris en compte.



Fig. 55 : Augmentation de la représentation du routing dans la partition traditionnelle

Quand cette représentation était totalement absente et que nos résultats indiquaient un changement, nous avons ajouté le même type de représentation.

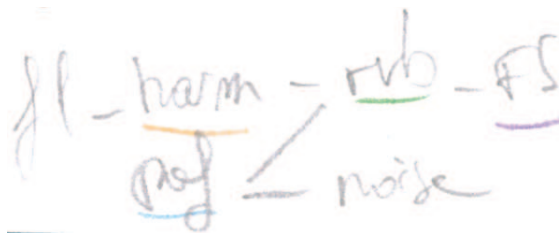


Fig. 56 : Représentation du Routing sur la partition traditionnelle depuis nos résultats

Chapitre IX

Analyse de *Jupiter*

Résumé

Ce chapitre est dédié à l'analyse de *Jupiter*. Nous l'effectuons en trois temps. Dans un premier nous présentons l'ensemble des éléments perceptifs qui permettent de saisir la pièce. Ces éléments ne sont pas du même ordre, certains sont attachés à des sonorités et d'autres expliquent des jeux de contraste entre sections. Afin de rendre compte de ces éléments nous représentons un *espace compositionnel* qui s'éclaire avec des *archétypes*, qui eux-mêmes englobent des *matériaux*. C'est au sein de cette structuration que nous éclairons ces éléments. Le deuxième temps analyse la pièce section par section, il s'agit d'expliquer ce que l'on entend avec la connaissance de l'artefact que l'on a acquis, et de témoigner de la construction de la pièce au travers des précédents éléments perceptifs. La troisième et dernière partie de cette analyse concerne la structuration de la pièce dans son entièreté. À l'aide des précédents *archétypes* nous apportons des premières considérations sur la structuration globale. Ensuite nous proposons d'éclairer la globalité de la pièce non à l'aide d'une forme cohérente, hautement structurée, mais par un vaste ensemble de réminiscences qui ont lieu tout au long des 37 minutes. Enfin nous apportons une représentation de ces réminiscences au sein du précédent espace compositionnel.

Note sur la segmentation

Dans la partition traditionnelle les sections se subdivisent en une seule sous-unité nommée *sous-section*. Elles sont repérées à l'aide de lettres majuscules, par exemple « IIC » indique la troisième sous-section de la deuxième section. Au fil de notre analyse nous nous sommes aperçus que ces sous-sections correspondaient moins à des niveaux structurels que des points de repère. Par exemple dans la section VI ces sous-sections, au nombre de 16, s'apparentent à des phrases tandis que dans la IV les sous-sections englobent des phrases, nettement scindées par des silences.

Pour mener à bien notre analyse nous avons donc utilisé une nouvelle segmentation en trois niveaux avec la terminologie suivante : une section se subdivise en *épisodes* qui se subdivisent en *phrases*.

Dès lors les sous-sections peuvent tantôt correspondre à des épisodes, tantôt à des phrases. Afin d'utiliser les repères de la partition nous avons simplement appliqué les règles de notation suivantes : lorsque les épisodes correspondent à des sous-sections nous les notons 1A, 1B, 1C ... ; lorsqu'ils englobent des sous-sections nous les notons 1A-C, 1D-E, 1F-I.

1 Espace compositionnel

1.1 Introduction

Pour mener à bien notre analyse nous présentons dans ce premier grand titre les éléments perceptifs les plus saillants de *Jupiter*. Tous ne sont pas du même ordre, certains relèvent simplement d'une sonorité et d'autres viennent éclairer un enchaînement d'épisodes ou de sections par exemple. Pour notre analyse nous avons différencié trois niveaux d'éléments.

Les premiers se remarquent essentiellement par leur caractéristique sonore, leur timbre. Il y a par exemple un intervalle de quarte augmentée, joué à la flûte, qui est récurrent tout au long de la pièce. Nous nommons ce premier niveau d'éléments le *matériau*¹⁹².

Le deuxième niveau s'éloigne du sonore pour se diriger vers les idées compositionnelles. Par exemple les importantes interpolations du Sampler ne sont que faiblement saisies par des descriptions sonores. Le Sampler peut en effet changer de sample, mais ce qui le caractérise le plus sont des amas d'impulsion qui se déplacent très amplement dans l'ambitus. Dans cette importante densité et dans ces mouvements d'ambitus il y a une (des) idée(s) compositionnelle(s). Ce deuxième niveau est dédié à ces idées musicales, nous les ciblons en parlant d'*archétype*.

Enfin, dans l'écoulement de la pièce nous percevons à plusieurs reprises des jeux d'opposition. La section V par exemple, avec les importantes interpolations du Sampler, s'inscrit en contraste de la IV, qui est « spacieuse ». Le troisième et dernier niveau nous permet de rendre compte de ces jeux de contrastes. Pour ce faire nous imaginons un large *espace compositionnel* dans lequel nous plaçons deux *zones protéiformes*, une qui relève de la *stabilité* et l'autre des *perturbations*. Ces dernières sont éclairées par les différents *archétypes*, qui eux même englobent le *matériau*.

1.2 Matériau

1.2.1 La phrase d'ouverture

La pièce s'ouvre avec une phrase de 8 notes jouées à la flûte en solo. Les notes de cette phrase sont déduites du nom de la personne à qui la pièce est dédiée, l'Arry BEAurEGArD.

192 Ces éléments correspondent au *son musical* du chapitre III (cf. 1.4)



Fig. 57 - Phrase d'ouverture

Il peut être tentant d'envisager cette phrase comme une série, qui ouvrirait une pièce sérielle. Mais cela n'est pas le cas puisque les intervalles présentés sont ici réutilisés tels quels. C'est une phrase qui se divise en cellules, et non une série sur laquelle effectuer des opérations de renversement et/ou rétrograde.

1.2.2 La cellule c

La phrase d'ouverture contient un intervalle clé qui est récurrent tout au long de la pièce, celui de quarte augmentée. Dans la phrase il est présent sur Lab₃ Ré₄, mais on le trouve le plus souvent transposé sur Do#₃ Sol₃. La pièce se termine sur cet intervalle.

1.2.3 Note polaire

L'installation de notes polaires, par simple répétition d'une note, est aussi récurrente. Ces notes sont jouées le plus souvent par la flûte et correspondent souvent à un Do#₃, comme dans la première section.

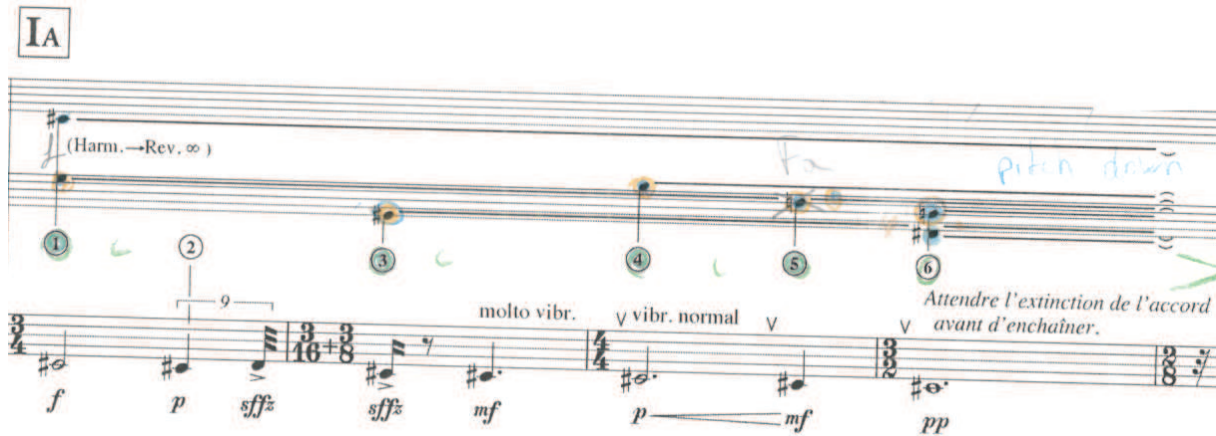


Fig. 58-Installation de notes polaires. Sous section IA

1.2.4 Motifs α et β

Dans les premières sections nous relevons deux motifs rythmiques joués à la flûte. Ils sont constitués de triples croches voisines et *staccato*.



Fig. 59 - Motif α (gauche) et β

Ces motifs s'apparentent volontiers à des interrupteurs de par le rendu machinal des triples croches *staccato* et par leur fonction : ils sont en effet utilisés pour marquer la fin d'une phrase, d'un jeu entre flûte et électronique.

1.2.5 Les triples de l'Harmonizer

Un jeu de questions-réponses entre la flûte et l'Harmonizer, avec des triples croches ascendantes, est aussi présent de façon régulière. Il est par exemple exposé dans l'épisode 1D.

Fig. 60 - Les triples de l'Harmonizer

1.2.6 Des agrégats

Plus fortement on retrouve tout au long de la pièce des agrégats. Ils permettent le plus souvent d'accompagner le jeu de la flûte en se juxtaposant sur ses notes. Ces agrégats peuvent être soit plaqués soit brièvement arpégés. Ils peuvent être produits de deux façons différentes, non exclusives. La première consiste à utiliser l'Harmonizer et la Reverb, la seconde le Paf.

La première méthode se fait avec la branche de routing suivante $fl \rightarrow Harm \rightarrow Reverb$. L'Harmonizer permet alors de transposer la note de la flûte sur un maximum de 4 hauteurs différentes et la Reverb de prolonger ces hauteurs *ad libitum*. Afin de se référer à cet usage très fréquent nous le nommons « système de transposition prolongation » et plus brièvement « le système TP ».

Le Paf peut être utilisé soit pour produire des agrégats à partir de rien, de les synthétiser, soit pour enrichir un agrégat déjà produit par le système TP. Le second cas reste le plus fréquent, le Paf reprend les notes transposées par l'Harmonizer et place des formants (8 maximum) au-dessus de leur fondamental.

Le début de la section II permet d'entendre ces deux méthodes d'obtention. Dès l'ouverture nous percevons un agrégat comprimé produit par le système TP, puis celui-ci est remplacé par un second produit par le Paf, mais aussi brouillé par le FS et le Noise.

The image shows a musical score for Section II A. The top staff is a flute line with routing '(Fl. -> Harm. -> Rev. ∞)'. The bottom staff is a piano accompaniment with dynamic markings (mf, p, pp, mf, ff) and tempo changes (rall., A tempo, Plus vite). Handwritten annotations include '1/5', '1', '3', '6', and 'r/b'.

Fig. 61 - Les deux moyens de production des agrégats, le système TP suivi du Paf

1.2.7 Brouiller des sons

Un dernier élément perceptif qui lie la flûte et l'électronique peut être décrit. Il s'agit d'utiliser le FS pour venir brouiller un son produit par l'électronique, souvent des agrégats. Ce brouillage est le plus souvent utilisé pour mettre en relief le jeu de la flûte : dans l'écoulement de ses notes, *fsfre* va recevoir une suite de valeurs afin de brouiller un agrégat. Les deux modes du FS peuvent être utilisés, soit le mode standard qui décale le spectre, soit le mode RM qui va ajouter des battements.

Fig. 62 - Les deux brouillages du FS

Dans l'exemple IF¹⁹³ nous retrouvons ces deux utilisations. Un agrégat est maintenu par la Reverb et le Paf puis, au cours du jeu de la flûte, le FS va venir le brouiller. Au tout début par décalage de spectre et ensuite en ajoutant des battements, et ainsi de suite.

1.3 Archétypes musicaux

Cinq archétypes nous permettent d'éclairer les grandes idées compositionnelles présentes dans *Jupiter*, nous les nommons *Aérien*, *Continuité*, *Interaction continue*, *Geste proéminent* et *Ostinato*. Ces archétypes se situent au-dessus des précédents matériaux, ils les englobent, et en conséquence leur description passe par la répartition de ces matériaux. En plus des matériaux que nous avons précédemment décrits il faut inclure ceux attachés à des Modules (e.g. le Sampler), décrits au chapitre VII.

1.3.1 Aérien

C'est un archétype très présent, facilement identifiable, qui permet de rendre compte entièrement de certaines sections – des sections dites « aériennes ». Ce qui le caractérise le plus est la formation d'un agrégat en début de phrase qui est maintenu tout au long. Cet agrégat est léger et aigu, éthéré, et comme il est plaqué sur une note de flûte il donne l'impression d'une « mise à l'horizon », d'un prolongement infini, une impression d'air, d'espace. Afin de différencier ces agrégats éthérés des autres, nous les nommons des *tapis*. Ensuite la flûte déroule un jeu lent et *piano* et/ou, par contraste, enchaîner une suite de triples croches, comme dans l'exemple ci-dessous.

193 Les valeurs de *fsfre* sont reportées sur la partition à l'aide de notes. Ces indications ouvrent d'importantes questions sur la thématique des représentations musicales puisque ces « notes » n'ont ici rien d'audible. Ces questions ne peuvent être traitées ici ; elles restent à approfondir dans un travail ultérieur.

Fig. 63 - Exemple d'une phrase type de l'archétype aérien

1.3.2 Continuité

C'est l'archétype qui englobe le plus de matériaux, de sonorités. Pour autant ceux-ci ne sont pas disparates mais s'inscrivent tous – et c'est ce qui caractérise le plus cet archétype – dans la « continuité de la flûte », c'est-à-dire de son timbre. Tous les matériaux que nous avons précédemment décrits peuvent ici être utilisés. Les agrégats sont par exemple très présents. Ils peuvent être arpégés ou plaqués, produits par le système TP et/ou par le Paf, mais tous sont en lien avec la flûte. Le jeu en questions/réponses entre flûte et Harmonizer est aussi présent. Le *brouillage* aussi, il va permettre justement de souligner le jeu de la flûte à l'aide des agrégats. Cet archétype est très diffus, il peut se retrouver localement dans plusieurs sections, sans pour autant en rendre compte. Il reste le plus présent dans les trois premières sections.

1.3.3 Interaction continue

C'est un archétype précis, facilement repérable, qui correspond à l'usage du Chapo. En plus du matériau produit par le Chapo – ce son de masse complexe sur lequel est apposée une enveloppe spectrale – cet archétype inclut l'idée musicale que l'algorithme permet de réaliser, à savoir le tissage de liens entre hauteur de note de flûte et évolution d'un son de masse complexe. La section VI développe *in extenso* cet archétype.

1.3.4 Geste proéminent

Cet archétype englobe la plupart des matériaux produits par le Sampler, ceux faits d'un amas d'impulsions qui se déplace amplement dans toute l'étendue du spectre. Nous qualifions ces derniers de *gestes proéminents* puisqu'ils sont en effet très présents. Relevons que la précédente distinction que nous avons faite entre interpolations et séquences n'est ici plus valide, puisque nous retrouvons ces gestes très amples dans les deux catégories.

Dans cet archétype nous distinguons deux sortes de gestes. Ceux qui respectent le principe de *neumes* du compositeur (cf. chapitre VII 1.1.3.c) et ceux au contraire qui ne le respectent pas, c'est-à-dire ceux dont les mouvements d'ambitus sont linéaires.

1.3.5 Ostinato

L'*ostinato* est aussi un archétype précis, facilement repérable. Il correspond aux *ostinatos* rythmiques, écrits soit pour la flûte soit pour le Sampler, soit pour les deux et avec un jeu de questions/réponses comme le présente la figure ci-dessous.

Fig. 64 - Jeu de questions/réponses avec l'archétype Ostinato

Cet archétype se retrouve de façon diffuse dans beaucoup de sections, mais il est surtout présent tout au long de la section VIII.

1.4 Deux zones protéiformes

Il y a dans *Jupiter* un jeu d'opposition assez frappant entre deux « mondes musicaux ». Évidemment par l'effectif de la pièce et par les intentions du compositeur¹⁹⁴ il est tentant de comprendre ces deux « mondes » comme celui de l'instrument traditionnel et celui de l'électronique, soit de penser l'esthétique à partir d'une musique mixte. Par exemple cette opposition semble pertinente dans les contrastes entre l'archétype *Geste proéminent* et celui *Aérien*, où le premier correspond à une écriture pléthorique au Sampler, avec un amas d'impulsions, que seul l'électronique peut restituer, et le second à une écriture dilatée à la flûte. Mais cette opposition est mise à mal à plusieurs reprises dans la demi-heure de la pièce. Plusieurs fois il y a en effet une écriture dense et machinale à la flûte, qui imite celle du Sampler (sous-section VIII-E par exemple) et, il y a aussi, comme nous l'avons vu, une écriture de l'électronique qui vient prolonger ou souligner le jeu de la flûte (e.g. les archétypes *Aérien* et *Continuité*), comme si « le monde instrumental » venait imiter celui électronique et inversement. Bien plus *qu'une* opposition il y a donc *des jeux* d'opposition, de va-et-vient. De plus, ces jeux nous paraissent mal envisagés dans l'opposition flûte ~ électronique ; ils se font au-delà des instruments, au-delà des producteurs sonores.

194 « Dans les trois dernières œuvres que j'ai produites à ce jour [Jupiter, Pluton, Partition du ciel et de l'enfer], mon principal but a été d'intégrer dans un même discours deux mondes qui, auparavant communiquaient de façon plus parallèle que conjointe : la musique instrumentale et la musique électronique ». [Manoury, 1998, p.43]

Pour rendre compte de ces jeux d'opposition nous proposons un espace compositionnel qui se dessine au travers de deux grandes zones. Une zone n'est pas à concevoir comme un cercle, fermé avec une limite, mais bien plus comme un groupement de plusieurs puits de lumière. Ces deux zones ne sont donc pas uniformes, mais elles se dessinent de part et d'autre de l'espace compositionnel sans qu'il y ait de régularité : elles ne possèdent pas un point d'intersection mais tout une continuité de points.

Ces zones, qui nous permettent de témoigner de l'esthétique de *Jupiter* de façon imagée, se dessinent dans l'ordre de description que nous avons donné des matériaux, de la « phrase introductive » aux « agrégats », et de même pour les archétypes, de l'*Aérien* à l'*Ostinato*. Dans ces enchaînements il y a de plus en plus de densité, de plus en plus d'excitation, de plus en plus de nervosité, et ainsi de suite. Afin de rendre compte des jeux d'opposition possibles, et en essayant de ne pas les limiter, nous qualifions la première zone de *stabilité* et la seconde de *perturbations*. Dans la première il y a les archétypes *Aérien*, *Continuité* et plus difficilement *Interaction continue*, puis dans la zone de *perturbations* il y a *Gestes proéminents* et fondamentalement *Ostinato*.

Nous proposons d'éclairer l'esthétique de *Jupiter* avec ces deux zones protéiformes. En substance, on pourrait dire que l'esthétique de la pièce se dessine au travers d'un agrégat plaqué aigu et maintenu, auquel succède une suite de triples croches *staccato* à la flûte.

2 Analyse

Section I

La pièce s'ouvre dans la zone de stabilité avec des notes polaires et des agrégats. Les phrases et les agrégats s'enchaînent, scindés par les premiers motifs rythmiques de la flûte. Au fil des 5 épisodes de la section cette stabilité va être bousculée, par itération, jusqu'à atteindre la zone de perturbation. Au dernier épisode la flûte joue en effet un *ostinato* et sur certaines de ses triples croches sont juxtaposés des éclats de la Paf, synthétiques et percussifs, comme des étincelles.

Comme nous l'avons vu la pièce commence avec la phrase d'ouverture jouée à la flûte seule, sans traitement ni synthèse. À cette fluidité s'oppose une première exposition du motif α avec ses triples croches *staccato*. L'épisode 1A (evt 1-12) poursuit cette alternance avec des notes polaires, des agrégats maintenus et un jeu de questions/réponses entre flûte et Harmonizer. Dans la première phrase (IA1) la flûte installe la note polaire Do#3. Elle est directement accompagnée par le système TP qui juxtapose un Sol2, et par le Paf qui double le Sol2 et le Do#3.

The image shows a musical score for Episode 1A. It consists of two staves. The top staff is a grand staff with a treble clef and a key signature of one sharp (F#). It contains a melodic line with six circled notes numbered 1 to 6. Above the staff, there are handwritten annotations: "(Ham. → Rev. ∞)", "p", and "piten down". The bottom staff is a piano part with a 3/4 time signature. It features dynamic markings: *f*, *p*, *sfz*, *mf*, *p*, *mf*, and *pp*. Performance instructions include "molto vibr.", "vibr. normal", and "Attendre l'extinction de l'accord avant d'enchaîner." There are also some handwritten notes like "9" and "10+8" above the piano part.

Fig. 65-Épisode 1A

Cette phrase se termine avec un glissement dans le grave du Paf, comme un signe des possibilités d'indépendance de l'électronique. La seconde phrase vient légèrement perturber cette stabilité avec un jeu de questions/réponses. La flûte ré-installe la note polaire Do#3 avec des croches espacées entre lesquelles l'Harmonizer, avec sa possibilité de délai, va placer des agrégats – soit plaqués soit arpégés. Cette légère perturbation s'éteint avec un agrégat comprimé qui sert de liaison avec le prochain épisode.

Le deuxième épisode (1B-C, evt 13-30) poursuit le même schéma en augmentant les perturbations avec l'apparition du FS. La première phrase commence avec la cellule c puis

installe, en parallèle du précédent agrégat, la note polaire Sol₃. La perturbation vient de l'Harmonizer et du FS. Le premier produit des triples croches et le second vient les brouiller en alternant d'importants décalages du spectre, de plus de 500 Hz, tantôt positifs tantôt négatifs.

Fig. 66 - Épisode 1B-C

La seconde phrase est une courte transition qui vient marquer la perturbation avec un *climax*. Ce dernier est amené par la flûte qui installe de nouveau la note polaire do₃, du *mezzo-forte* au *fortissimo*, puis par l'Harmonizer avec une montée en triples croches jusqu'à atteindre le sur-aigu, La6.

Le troisième épisode (1D, evt 31-46) amène une nouvelle perturbation à l'aide des triples de l'Harmonizer, situées dans le sur-aigu, qui s'éloignent du timbre de la flûte. L'épisode est construit avec un jeu de questions/réponses entre flûte et Harmonizer. Le contraste avec les précédents épisodes vient de la flûte qui cesse d'installer des notes polaires pour jouer des triples croches dans le medium aigu, et surtout de l'accentuation de certaines croches de l'Harmonizer, par le Paf et la Reverb, qui vont produire des sonorités synthétiques dans le suraigu.

Fig. 67 - Épisode 1D

Les accents du Paf sont produits en plaçant des formants entre 6 et 15 demi-tons au-dessus des notes de l'Harmonizer. La Reverb de son côté capture et prolonge ces mêmes croches de sorte à former, au fil de l'épisode, un nouvel agrégat. Cette capture¹⁹⁵ se fait en maniant temps de délai, ouverture, puis fermeture de la porte de la Reverb (*rgate*), comme le montre l'extrait des qlist¹⁹⁶ :

```
trans1 500;
125 trans2 1100;
125 amp2 110 5;
rgate 117;
revfb 127;
trans3 2200;
125 trans4 400;
rgate 0;
```

Le quatrième épisode (1E-F, evt 46-87) retourne dans la zone de stabilité. Il commence avec une courte phrase de flûte solo, sans traitement, et se termine avec la cellule c, qui sera maintenue par le Paf et la Reverb sur tout le prochain épisode. Ces deux écritures, présentes dans IA, marquent une sensation de recommencement. La seconde phrase expose un nouveau jeu qui vient développer cette zone de stabilité. La flûte a un jeu expressif, construit autour de la note polaire Do#₃, fait d'un large ambitus (Do#₃-Sol₅), de nuances, de liaisons et d'appoggiatures.

195 La métaphore de la chauve-souris du compositeur est ici très parlante (cf. Chapitre VII 2.3.4)

196 La noire étant à 60, une triple croche correspond, machinalement, à 125 ms. Il faut avoir en tête que les retards se cumulent : *amp2* et *trans3* arrive 250 ms après *trans1* : ils engendrent donc la troisième triple croche.



Fig. 68 - Épisode 1E-F

Comme le montre la figure ce jeu est augmenté par le FS qui va brouiller le précédent agrégat. Sur les Do#3 le FS ajoute des battements (mode RM) et au fil des notes de la flûte le FS apporte des décalages.

Le dernier épisode (1G, evt 88-108) se situe pleinement dans la zone de perturbation. Il ne s'agit plus de bousculer la stabilité, comme en ID, mais de donner un premier aperçu de cette zone faite de perturbations. Deux nouveaux matériaux sont exposés : l'ostinato machinal réalisé à la flûte et les éclats de la Paf, percussifs et saillants, qui font écho aux accentuations plus timides rencontrées en ID.

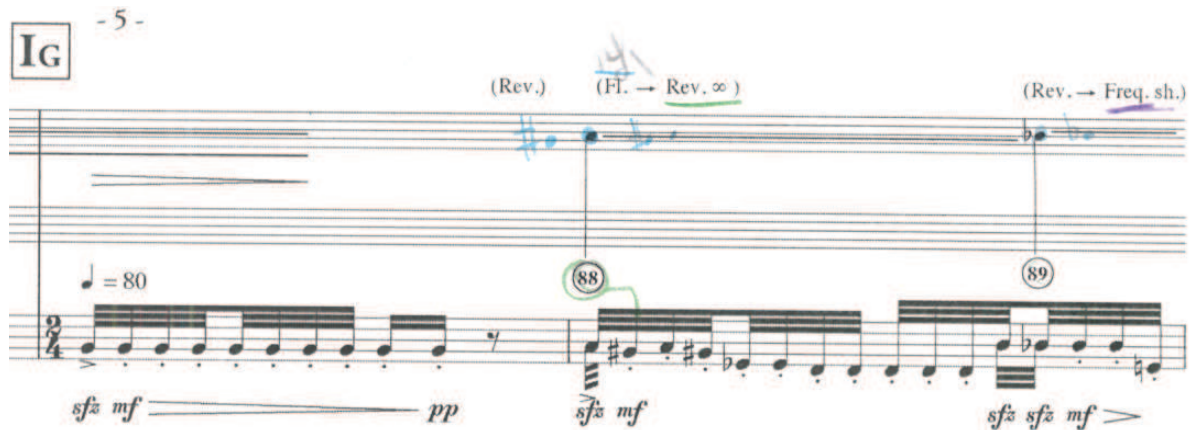


Fig. 69 - Épisode 1G. Les éclats du Paf

Les éclats sont obtenus par le Paf en combinant (i) des enveloppes d'amplitude franches et brèves¹⁹⁷, avec (ii) des formants placés très au-dessus (quasi 3 octaves) des notes de la flûte, comme le présentent les lignes suivantes de l'événement 88.

- 3 La3 cf +35 bw 43
- 4 Sol#3 cf +31 bw 40

¹⁹⁷ 5 ms d'attaque et de chute et 30 ms de maintien (cf. Chapitre VII 1.4).

De plus, 60 ms après leur extinction, ces éclats sont prolongés par le Paf et la Reverb ; ils sont alors placés dans la même octave que celle de la flûte.

Section II

La deuxième section utilise aussi l'archétype *continuité* pour éclairer une nouvelle fois la zone de *stabilité*. Nous retrouvons ainsi le long de ses 3 épisodes le même matériau que précédemment. Le premier ré-expose celui de la flûte, expressif, le deuxième apporte une perturbation avec une multitude de triples croches de l'Harmonizer, et le dernier expose la limite de la stabilité avec un rythme soutenu à la flûte. Enfin, l'épisode se termine avec une *codetta* aérienne.

Le premier épisode (2A, evt 1-50) éclaire plus amplement la zone de stabilité. La flûte est au premier plan et l'électronique l'accompagne en mettant en relief son jeu. La première phrase de l'épisode peut être comprise comme un condensé partiel de la section.

Fig. 70 - Épisode 2A

En premier il y a la flûte avec un jeu fluide et lié qui rappelle la phrase d'ouverture ; en second il y a l'enchaînement de deux agrégats, plaqués sur le jeu de la flûte. Le premier est produit par le système TP, il a une sonorité très proche de la flûte, telle une « mise à l'horizon », à l'inverse le second est produit par le Paf et le Noise, son rendu est synthétique, haché, il vient troubler la stabilité.

La deuxième phrase est une courte transition qui fait écho à l'épisode IE-F. On retrouve en effet le jeu expressif de la flûte avec des appoggiatures et les brouillages du FS. La troisième phrase approfondit l'expressivité avec un jeu plus rapide à la flûte fait de triples croches liées. Elle est construite sur une alternance entre des passages *forte*, vifs et expressifs, et des passages

piano, faits de croches *staccato*. Ces variations sont appuyées par des agrégats arpégés lors des passages *piano*, et pour les passages *forte* l'ajout de battements.

Fig. 71 - Épisode 2A

Le deuxième épisode (2B, evt 51-58) vient perturber cette stabilité avec le même matériau que dans l'épisode ID, un jeu de questions/réponses entre flûte et Harmonizer. La flûte joue des triples croches ascendantes qui vont lancer les réponses de l'Harmonizer. Certaines des triples de l'Harmonizer sont accentuées par le Paf, avec des éclats qui rappellent ceux de l'épisode IG, mais ici nettement moins percussifs, saillants, puisque le temps qui sépare leurs attaques et leurs maintiens (60 ms auparavant) est maintenant nul.

Le dernier épisode (2C, evt 59-81) se situe à la croisée des deux précédents. La flûte est au premier plan avec un jeu expressif qui reprend celui du premier épisode mais ici de façon plus marquée : le rythme est plus soutenu et les nuances s'enchaînent.

Fig. 72 - Épisode 2C

De même, l'électronique souligne le jeu vif de la flûte avec plus de relief. Des agrégats sont formés par le système TP et doublé par le Paf sur les notes accentuées de la flûte, et le FS est utilisé pour ajouter du relief. La dernière accentuation (evt 74) est la plus saillante. Elle est

obtenue en plaçant les formants du Paf dans le suraigu (une octave plus 9 demi-tons au-dessus du Sol#5 prescrit) et avec le FS qui ajoute de rapides battements.

La section se termine avec une *codetta* qui vient rompre ce rythme soutenu avec un rendu très aérien, éthéré, où les notes de flûte sont « projetées dans l'horizon ». Elle expose l'endroit le plus stable de la stabilité, là où le temps s'écoule sans qu'on puisse le remarquer. Ce rendu est obtenu avec le Sampler qui installe à deux reprises des notes polaires. Cette répétition est augmentée par l'Harmonizer qui applique un délai sur les « notes » du Sampler, créant ainsi une sorte de flou temporel.

Section III

Cette section développe considérablement la zone de stabilité qui s'était dessinée au fil des dernières sections. Elle nous montre au travers de 4 épisodes d'autres endroits avec par exemple un usage abondant du Paf, qui se substitue au système TP, ou encore le matériau *ad hoc* des *vaguelettes*. Le premier épisode est une synthèse des précédents jeux rencontrés à la nuance que nous sommes « loin du timbre de la flûte » ; le deuxième est une transition qui fait écho à l'épisode ID ; le troisième expose les *vaguelettes* ; le dernier à l'inverse poursuit la voie du 2C et pousse les limites de la stabilité avec un jeu saccadé entre flûte et agrégats.

Le premier épisode (3A-D) relance un jeu qui se situe dans la stabilité mais nettement plus loin du timbre de la flûte qu'auparavant. Les agrégats sont en effet produits intégralement par le Paf, ce qui les rend plus percussifs, plus présents. La première phrase expose ce nouveau matériau avec un jeu expressif à la flûte, marqué par des agrégats *fortissimo* très présents. Ces derniers ne sont pas fixes, continus, comme l'étaient ceux du système TP, mais ici mobiles grâce à deux effets. Le Noise d'abord. Il est appliqué sur chacune des voies du Paf avec un fondu d'entrée¹⁹⁸. Le Spat ensuite qui applique un effet de tremolo spatialisé¹⁹⁹, récurrent par la suite. La phrase se finit avec une exposition des deux motifs α et β .

Les deux phrases suivantes s'inscrivent en contraste, une perturbation suivie d'un passage très stable. La perturbation est obtenue à la flûte avec des croches aiguës *fortissimo* sur lesquelles le Paf plaque des agrégats percussifs et étendus.

198 Dès l'apparition de l'agrégat la connexion du Paf au Noise s'active en 200 ms (*noisetutti o, 127 200*).

199 *spatinc = 4*

Fig. 73 - début de la deuxième phrase de 3A-D

La présence des agrégats est augmentée avec le FS qui les décale dans le grave, et d'autre part avec le Spat qui applique un tremolo spatialisé très rapide²⁰⁰. C'est ce dernier effet qui annonce la stabilité, avec un décroissement en vitesse, un *rallentando* (*spatinc* passe de 127 à 4 en une seconde). La stabilité est ensuite construite avec des tremolos *pianissimo* à la flûte, qui se fondent dans des agrégats lisses, avec de longues attaques et sans brouillage. La dernière phrase vient mettre fin à ce jeu avec un grand geste descendant. La flûte joue des croches accentuées sur lesquelles sont plaqués des agrégats qui se dirigent vers le grave. L'effet de fin est augmenté par le tremolo spatialisé qui est appliqué dans la plupart des résonances des agrégats (*spatinc* passe de ~110 à 4 en 1 sec.). Le motif β met fin à ce déclin.

Le deuxième épisode (3E, evt 34-63) est une courte perturbation très marquée qui nous amène à la limite de la stabilité. Il commence par une brève note polaire au Sampler, sur La₄, à laquelle succède un jeu très vif à la flûte accentué par les éclats du Paf. Ce jeu nous rappelle l'épisode 1D, mais il se trouve ici plus marqué, telle une ré-exposition qui va plus loin. Les éclats du Paf s'apparentent en effet à des sons de cloches, très cristallins, situés dans le sur-aigu.

Le troisième épisode (3F, evt 65-100) expose un matériau singulier qui éclaire d'une nouvelle façon cette vaste zone de la stabilité. Il s'agit des vaguelettes électroniques, produites par le Sampler et/ou par Additive (Cf. Chapitre VII 1.2.3). Ces vaguelettes sont des petits gestes très fluides, composés d'une dizaine d'impulsions soient ascendantes soit descendantes. Comme nous l'avons vu, la fluidité de ces vaguelettes vient d'une part du maintien des sinusoïdes d'Additive d'un geste à l'autre, et d'autre part d'un temps de réverbération conséquent (*revfb* >

²⁰⁰ *spatinc* = 127

100), où les sinusoïdes de l'un et les samples de l'autre viennent se fondre. Entre les vaguelettes il n'y a donc pas de silence mais le maintien d'un matériau léger et lointain. L'image que nous avons de ce matériau est celui d'un environnement très élastique, fermé, où une perturbation à un endroit localisé vient se répercuter à un autre endroit, et ainsi de suite. Cet épisode singulier est fait de trois phrases sous la forme exposition, développement et conclusion.

L'épisode commence avec des vaguelettes qui se juxtaposent à un jeu très libre à la flûte, « *libre, étendu, quasi improvisando* » peut-on lire sur la partition²⁰¹. Les vaguelettes sont très homogènes, produites par le Sampler et par Additive. Pour le développement le matériau est plus riche, plus contrasté. La flûte enrichit son jeu avec des triples croches *staccato* et installe la cellule c transposée sur Ré₄ Sib₄. Les vaguelettes sont aussi plus contrastées, certaines sont percussives et d'autres disparaissent subitement.

The image shows a musical score for the development of episode 3F. It consists of five staves. The top staff is a treble clef with complex rhythmic patterns and dynamics like *mp* and *p*. The second staff is a treble clef with a melodic line and dynamics like *mp* and *p*. The third staff is a treble clef with a melodic line and dynamics like *ff* and *mf*. The fourth staff is a bass clef with a melodic line and dynamics like *ff* and *pp*. The fifth staff is a treble clef with a melodic line and dynamics like *f* and *mf > p*. There are several handwritten annotations in blue and green ink, including 'syn', 'open', and '3'. A circled number '79' is in the fourth staff, and circled numbers '80', '81', and '82' are in the fifth staff. At the bottom, there is a line of text: 'Attendre l'extinction de la partie synthétique avant d'enchaîner.' and a signature 'Spatine'.

Fig. 74 - Développement de l'épisode 3F

Comme le montre la figure, l'effet de stabilité de cet épisode est augmenté par le Sampler qui installe des notes polaires, en répétant localement certaines des notes de la flûte (evt 79).

La dernière phrase conclusive se fait en deux temps, une phase calme suivie d'un *climax*. Il y a d'abord des vaguelettes, très fluides, produites par Additive, auxquelles se juxtapose un jeu *pianississimo* à la flûte. Par contraste il y a ensuite le *climax*, un important mouvement vers

²⁰¹ Toutes les indications textuelles de la partition sont reportées dans l'annexe F.

l'aigu réalisé par le Sampler et la flûte, toujours de façon fluide comme en témoigne la figure ci-dessous.

The image shows a musical score for the climax of episode 3F. It consists of five staves. The top two staves are for the flute and sampler, with the flute part featuring a series of triplets of eighth notes. The bottom three staves are for the piano, with the left hand playing a rhythmic pattern of eighth notes and the right hand playing a series of chords. The score is marked with dynamics such as *ff* and *pp*, and includes a crescendo and decrescendo hairpin. The measures are numbered 96, 97, 98, 99, and 100. There are handwritten annotations in red and blue ink, including the word 'climax' at the top and some notes in red.

Fig. 75 - Climax de l'épisode 3F

Le dernier épisode (3G, evt 100-150) vient perturber cette stabilité. À l'instar de l'épisode 2C nous nous trouvons ici à une extrémité de la stabilité, mais de façon plus marquée. Cet épisode est fait d'une seule phrase menée par la flûte. Son jeu est construit à partir de trois intervalles clés autour desquels elle va dérouler des triples croches abondamment nuancées. Il y a d'abord un intervalle de quarte (Si₃, Mi₄), ensuite une seconde majeure (Fa₄ Mib₄) puis de nouveau une quarte (Fa₃ Sib₃). Le Paf accompagne la flûte avec deux écritures. Il accentue certaines des triples croches de la flûte de sorte à mettre en relief les précédents intervalles, puis, dans les courts temps de silences de la flûte le Paf vient plaquer des agrégats par a-coups de sorte à casser toute continuité.

Section IV

Cette section nous plonge dans la stabilité avec un matériau qui sera récurrent dans la longueur de la pièce, celui de l'archétype *aérien*. Des fragments de cette zone ont été précédemment éclairés (e.g. la *codetta* de la section II, ou encore la phrase 3C) et il s'agit maintenant de la présenter une première fois. La section est composée de deux épisodes qui respectent la même organisation. Il y a au début formation d'un tapis, situé dans l'aigu et

produit par le système TP, et ensuite déroulement d'un jeu lent et *piano* à la flûte, souvent contrasté par des triples croches aiguës et accentuées.

Le premier épisode (4A, evt 1-16) commence par la formation d'un tapis dans l'aigu, très épuré puisque produit uniquement par la Reverb. Il est ensuite contrasté par un ensemble de triples croches *fortissimo* à la flûte.

Fig. 76 - Début de l'épisode 4A

L'épisode se poursuit avec un autre tapis, similaire, formé uniquement par la Reverb, sur lequel la flûte déroule lentement une série de *flutterzunge pianissimo*, caractéristique aussi de l'archétype.

Le second épisode (4B, evt 17-25) apporte une nouvelle lumière. Les tapis sont maintenant formés par le système TP, moins aigus et plus riches, et la flûte développe un jeu lisse dans le medium, sans triples croches. Au milieu de ces tapis entrecoupés de silence, une phrase se démarque amplement. Après la formation d'un deuxième tapis nous pouvons lire sur la partition « *baisser l'amplification de la flûte* ». Nous entendons alors le son de la flûte passé au travers de l'Harmonizer qui est en mode flouteur sur ces quatre voies. Le jeu de la flûte qui suit est calme, situé dans le medium-grave, il se démarque avec des *glissandos* descendant sur des tierces mineures. Cette continuité va ensuite être rompue subrepticement par deux croches aiguës et stridentes à la flûte, qui amènent le dernier tapis. La stabilité revient dans le déroulement de la flûte, avec des noires *flutterzunge*.

Section V

Cette section présente un dialogue entre la flûte et le Sampler qui nous amène dans la zone de la perturbation : la flûte est au second plan et le Sampler expose des gestes proéminents (sample de piano) qui se déplacent amplement dans l'ambitus. Ce passage clé est d'abord

précédé d'un épisode « de mise en marche », où les gestes proéminents prennent lentement forme. Il est suivi d'une rencontre de la flûte et du Sampler, par alternance, ce qui nous dirige vers la stabilité.

Le premier épisode (5A, evt 1-6) ouvre le dialogue. La flûte commence avec un long tremolo qu'elle répétera, entrecoupé de silences, le long de l'épisode. Le Sampler commence dans le grave au lointain avec un rendu peu discernable puis, au fil de l'épisode, il va être de plus en plus présent, comme si quelque chose prenait forme.

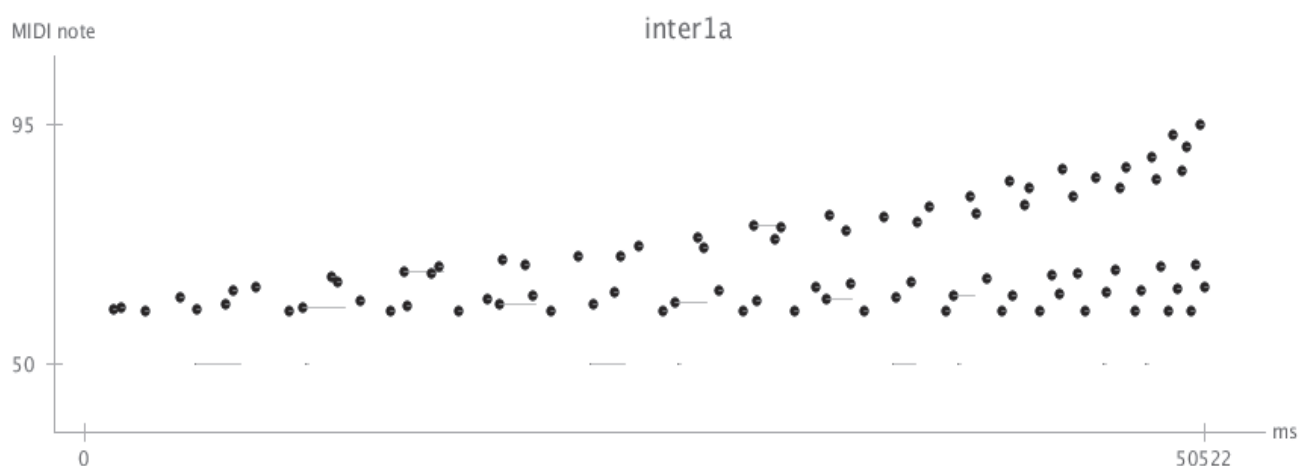


Fig. 77 - Geste proéminent du Sampler, épisode 5A

Comme le montre la figure²⁰² cette impression « de mise en route » se retrouve sur le plan des hauteurs et sur celui du rythme. Au début les impulsions sont à la fois espacées et désordonnées, puis, petit à petit, elles se dirigent vers le médium et vont être plus nombreuses. La progression n'est pas linéaire : elle se fait avec les « neumes » du compositeur, une impulsion aiguë est suivie de quelques autres plus graves et ainsi de suite. C'est cet effet qui donne l'impression d'une mise en marche laborieuse. En arrière plan se trouve un agrégat formé des deux dernières notes de la section précédente, la cellule c transposée sur Fa₃. Ces trois éléments, la flûte, le Sampler et en arrière plan l'agrégat, restent ici indépendants les uns des autres²⁰³.

Le deuxième épisode (5B, evt 7-11) « laisse la parole » au Sampler pour nous amener plus loin dans la zone de perturbation. Il expose un jeu très dense fait d'un amas d'impulsions, propre à l'archétype *geste proéminent*. Comme le montre la figure suivante, l'épisode est fait de plusieurs déplacements très amples en ambitus.

²⁰² Voir Chapitre VII 1.1.3.f pour l'explication des représentations du Sampler.

²⁰³ Ainsi peut-on lire sous la portée du flûtiste « Jouer les événements 1 à 5 *ad lib*. Pendant la durée de la séquence (approx. 40''). ».

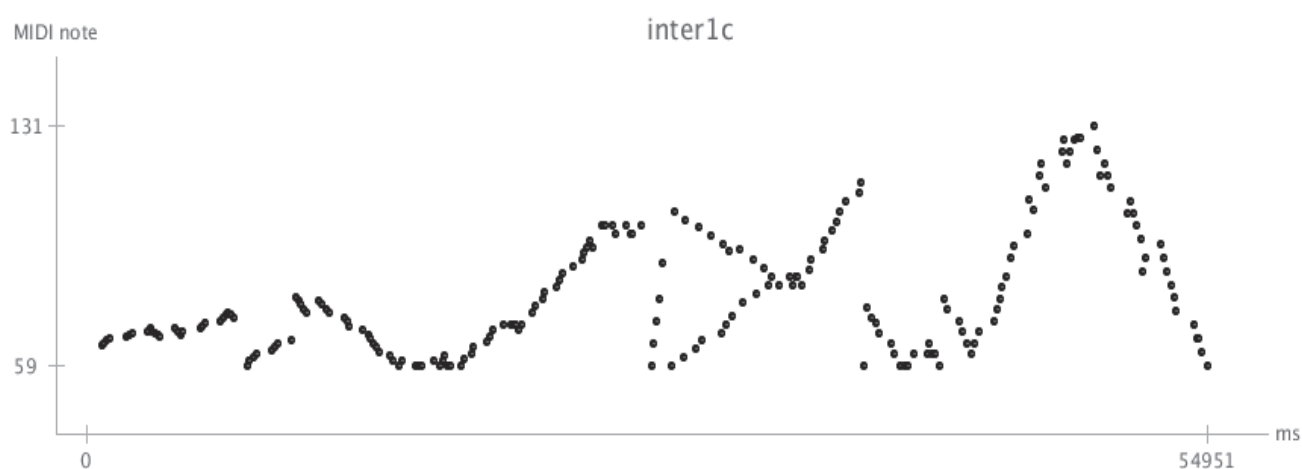


Fig. 78 - Geste proéminent Sampler, épisode 5B

La proéminence du Sampler est augmentée par l'Harmonizer qui applique, différemment sur chacune des voies, une transposition inférieure à deux tons et un retard de l'ordre de 300 ms.

Le dernier épisode (5C, evt 12-17) présente la rencontre de la flûte et du Sampler. Alors que précédemment ces protagonistes fonctionnaient indépendamment, ils avancent maintenant ensemble. Lorsque la flûte joue la note polaire Do_3 le Sampler lui répond et lorsqu'elle quitte cette note pour des croches plus aiguës, le Sampler s'arrête et ainsi de suite sur la durée de l'épisode. Cette réunion se retrouve dans le routing, où la branche Harmonizer $\rightarrow FS^{204}$ est appliquée sur la flûte et le Sampler. Le rendu du Sampler est similaire au premier épisode, il n'est pas fait de déplacements importants dans l'ambitus mais reste dans le registre medium-grave.

L'épisode se termine avec une *coda* qui nous replace dans la zone de stabilité, éclairée par l'archétype *aérien*. La flûte joue dans le medium *piano* et la Reverb prolonge un tapis fin, fait d'une seule note aiguë, Fa_5 .

Section VI

Cette section présente une zone atypique située à l'intersection de la stabilité et des perturbations. Au centre de la section il y a un jeu entre la flûte et le Chapo, qui s'établit à l'échelle des phrases. La section est faite de 16 phrases et il est possible d'induire un certain modèle qui permet d'expliquer d'une part leur organisation et de l'autre l'écriture de la section.

²⁰⁴ L'Harmonizer est en mode flouteur et le FS en mode RM.

Il s'agit d'un modèle tensions/détentes fait de deux phases. La première commence par la note polaire Do₃ à la flûte. Située quasi à la limite inférieure de la tessiture ce son est fortement caractéristique de par l'instrument qui le joue et par l'écriture qui en est faite. Le grave de la flûte amène un son avec une longue attaque, de l'ordre de 50 ms, durant lequel le spectre s'établit progressivement (voir figure ci-dessous), ce qui entraîne une impression forte de *commencement*. Cet effet est de plus augmenté par un point d'orgue sur chacune de ces notes.

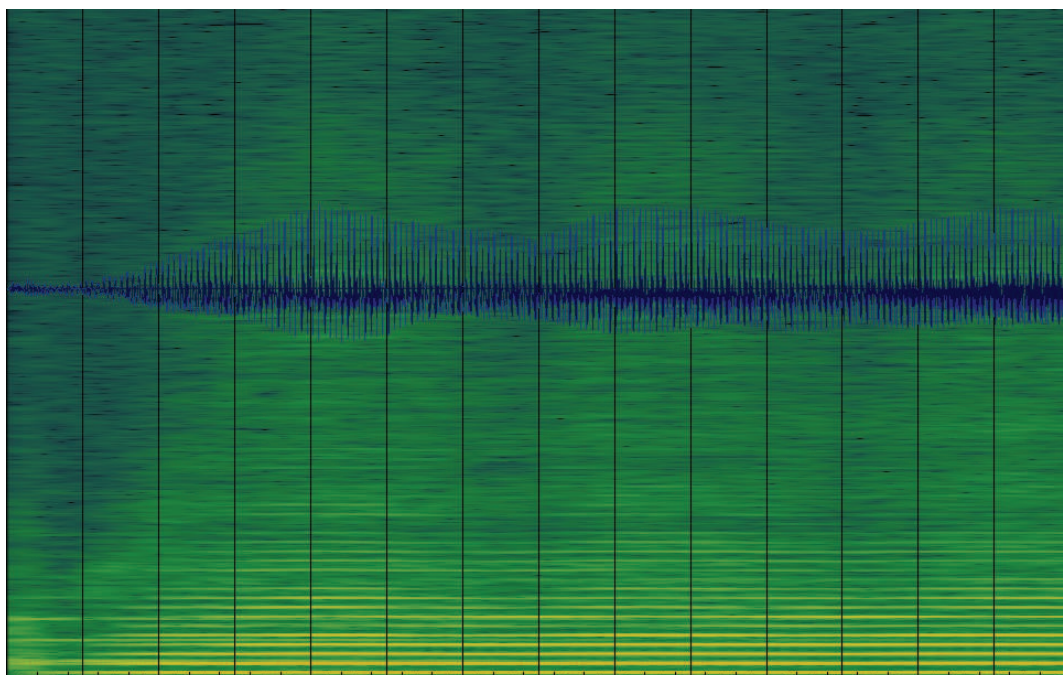


Fig. 79 - Forme d'onde et spectre du Do₃ de la flûte
Les lignes noires verticales représentent 50 ms
Fenêtre de Hanning, 2048 échantillons

Fig. 80 - Phrase type du modèle tensions/détentes

Par contraste, le Chapo juxtapose sur cette note d'assise un impact de masse complexe, percussif²⁰⁵ et inharmonique. Comme nous l'avons vu cet impact est le résultat de l'application d'une enveloppe spectrale sur un stack de fréquences²⁰⁶, et l'originalité de l'algorithme du Chapo est de faire déplacer cette enveloppe en fonction de la hauteur de la note jouée par la flûte, avec la correspondance suivante : « plus la note est aiguë, plus l'enveloppe se place dans le haut du spectre (donc plus elle est présente) et inversement ». Les impacts ne sont donc pas égaux, ils dépendent de l'importance de l'intervalle qui initie la phrase. Plus la dernière note de la précédente phrase est aiguë, plus l'impact sera marqué. Aussi, sur le long du point d'orgue des notes d'assises – de l'ordre de la seconde – l'impact évolue : un balayement est effectué du haut du spectre, riche avec beaucoup d'énergie, vers le bas avec nettement moins d'énergie.

La deuxième phase du modèle ne peut être décrite aussi précisément car les variations entre les phrases sont nombreuses. Néanmoins, le comportement suivant peut aisément être observé : à l'issue de la note d'assise la flûte enchaîne des notes plus courtes (de l'ordre de la noire) et plus aiguës, de sorte que plus la phrase s'écoule, plus les notes sont aiguës. Ces changements de hauteurs font évoluer l'enveloppe spectrale du Chapo, soit de façon instantanée, soit de façon fluide avec un certain lissage de l'enveloppe, selon la variable *porta*.

Nous pouvons ainsi éclairer l'organisation des 16 phrases de la section. Le commencement est marqué par un impact de masse complexe, percussif, contrasté par le Do₃ de la flûte en point d'orgue. Durant ce maintien, l'impact s'amointrit progressivement en amplitude et en hauteur. De la tension est ensuite amenée par la flûte avec des notes de plus en plus aiguës, jusqu'à parfois atteindre le sur-aigu afin de former des climax (e.g. evt 21). La section se compose de deux épisodes et d'une coda.

Le premier épisode (6A-F, evt 6-13) est une exposition de ce jeu de tensions/détentes entre flûte et Chapo. Il commence par une ouverture avec un geste très calme, une longue attaque du Chapo sur laquelle la flûte appose la note d'assise. Une tension ensuite se fait sentir avec le déplacement de l'enveloppe du Chapo dans l'aigu. L'ouverture se poursuit avec la phrase B qui respecte le précédent modèle. La détente est ici augmentée par la flûte qui joue un motif rythmique en *rallentando*. Par contraste l'ouverture se termine avec une pointe de tension où la flûte se synchronise sur le Chapo pour jouer un impact bref et percussif, le seul de la section.

L'épisode se poursuit avec une tension très marquée qui annonce un développement. L'impact est maintenant de forte amplitude, il correspond à un intervalle de 25 demis-tons, et de

205 Parmi les 16 phrases deux ne commencent pas par un impact percussif : elles marquent l'ouverture des épisodes.

206 Voir Chapitre VII.3. Le terme *stack* désigne un empilement de fréquences.

plus son spectre se densifie avec un doublement du nombre de composantes sinusoïdales²⁰⁷. Le développement se poursuit selon le modèle tensions/détentes. Deux effets viennent colorer ce jeu : (i) le FS en mode RM appliqué au Chapo qui va ajouter localement des battements (evt 6 et 12) ; (ii) des déplacements instantanés de l'enveloppe spectrale selon les notes de la flûte (la variable *porta*)

The image shows a musical score for three staves. The top staff is a treble clef with handwritten notes in blue and green, including 'H-harm = har' and 'chose' and 'noix - rub'. The middle staff is a bass clef with handwritten notes in purple, including 'RM 12 H' and 'FS P'. The bottom staff is a treble clef with handwritten notes in purple, including '6', '7', and '8'. The score includes performance markings such as 'mf', 'ff', 'p', and 'pp', and tempo markings like '♩ = 60', '♩ = 72', and 'rall.'. There are also dynamic markings like '3' and '3' over some notes. A box labeled 'VID' is present above the middle staff. The score is divided into measures, with some measures containing rests and others containing notes.

Fig. 81 - Extrait de l'épisode 6A-F

L'épisode se termine avec un geste qui nous dirige pleinement dans la stabilité. L'impact est harmonique, bâti sur La_3 , et la flûte prolonge la note d'assise Do_3 , suivie des notes consonantes Mi_3 et La_3 , auxquelles sont ajoutés des légers battements.

Le deuxième épisode (6G-P, evt 13-29) expose plus amplement ce jeu de tensions/détentes. Il est fait d'une ouverture (phrases G-H), d'une importante phase de stabilité (phrases I-K) et d'une phase de décroissance (phrases L-P).

L'épisode commence doucement avec la juxtaposition de deux attaques longues, celle du Chapo et celle de la flûte avec la note d'assise. Une pointe de tension est ensuite amenée par la flûte qui répète deux intervalles de quarte, auquel répond le premier impact percussif. Cette ouverture se termine calmement avec une trille. La suite de l'épisode se démarque avec une phase plus stable où ce jeu de tensions/détentes s'estompe durant trois phrases. Durant celles-ci la flûte enchaîne trois notes polaires situées dans le grave (Sol_3 , Do_3 et $Ré_3$) et conséquemment le spectre du Chapo reste aussi dans le grave. Cette stabilité est colorée à plusieurs reprises avec un intervalle à la flûte de sixte mineure, que le Chapo reflète avec des mouvements synchrones de l'enveloppe spectrale.

²⁰⁷ Les tables des oscillateurs ont maintenant deux composantes, un fondamental plus un harmonique de rang 3.

Fig. 82 - Épisode 6G-P

Cette phase de stabilité s'éteint avec le climax de la section. Comme le montre la figure il est amené par la flûte avec des triples croches qui mènent à Mi₅, la note la plus haute de la section. Ce geste engendre d'abord un déplacement vers l'aigu très saillant de l'enveloppe spectrale et ensuite un impact de forte amplitude, qui forme le climax.

L'épisode se termine avec un mouvement de décroissance sur 5 phrases. Il est encadré d'un geste chromatique à la flûte, fait de doubles croches *rallentando*. La première exposition de ce geste met fin au *climax* et annonce la décroissance : les impacts sont de moindre amplitude ; la flûte n'a plus d'envolée dans le suraigu et elle joue la note la plus grave, Si₂, qui amène une sensation de fin. La seconde expression du geste chromatique amène une résolution harmonique, fortement consonante (l'impact est harmonique bâti sur Sol₂ et la flûte joue Sol₃). Ce jeu de tensions/détentes avec la flûte et le Chapeau se termine donc sereinement en rythme, en nuance et en harmonie.

La *coda* est un jeu de questions/réponses entre flûte et Chapeau. Il vient révéler à l'auditeur le lien qui a structuré cette section, entre la hauteur des notes de la flûte et les balayements de l'enveloppe spectrale. La flûte joue des notes espacées avec de grands intervalles (de Do₃ à La₅), qui couvrent quasiment sa tessiture, et à cela correspondent des déplacements très amples et très vifs de l'enveloppe spectrale.

Section VII

Cette section courte s'inscrit pleinement dans la zone de stabilité avec l'archétype *aérien*. Elle fait ainsi écho à la IV, où des tapis aigus sont présents sur la longueur des phrases. Cette section se démarque cependant par sa limpidité : elle est faite de trois phrases, bien scindées, qui possèdent la même organisation. Ainsi nous pouvons voir cette section comme le prolongement de l'éclaircie amenée par la IV ; comme une seconde projection, plus affinée.

La section s'ouvre avec trois doubles croches aiguës à la flûte, stridentes, qui par contraste viennent former le tapis aérien. Ces doubles croches sont en effet prolongées par la Reverb et doublées par le Paf. Dans le second temps de la phrase – et non sans lien avec le précédent modèle – la flûte amène une ligne de tension avec un jeu rythmique fait de doubles et triples croches²⁰⁸.

Fig. 83 - Deuxième phrase de la section VII

La section se poursuit avec cette écriture sur les deux phrases suivantes. Elle se finit par contraste avec une *codetta* faite de triples croches accentuées qui fait la transition avec la prochaine section, dédiée à l'*ostinato*.

Section VIII

Par contraste cette section s'inscrit pleinement dans la perturbation. Elle expose en effet le matériau de l'archétype *ostinato* fait d'un amas de triples croches, tantôt au Sampler avec le sample percussif de Tam Tam, tantôt à la flûte. La section intègre trois épisodes. Il y a d'abord l'installation de l'*ostinato* puis, dans un temps plus conséquent, son développement et enfin son déclin.

²⁰⁸ Il est prévu que ces motifs soient enregistrés et utilisés pour alimenter les interpolations de la section IX.

L'épisode (8A-C, evt 1-7) nous place tout de suite dans le cœur de l'*ostinato* avec un rendu machinal à la flûte comme au Sampler. Au début il y a une phrase de questions/réponses entre ces derniers, de sorte qu'une dizaine de triples croches de l'un répondent à une dizaine de triples de l'autre et ainsi de suite. La flûte imite le jeu machinal du Sampler.

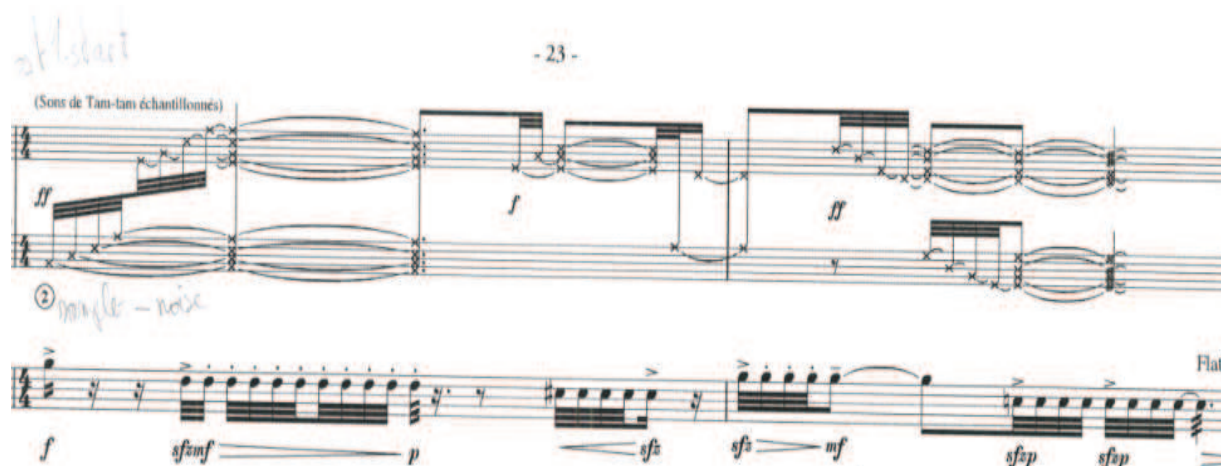


Fig. 84 - extrait épisode 8A-C

La flûte ensuite se juxtapose au Sampler avec *a contrario* un jeu léger, fait d'un tas d'appoggiatures, qui nous rappelle celui de l'épisode IE-F. L'épisode se termine dans l'*ostinato* avec le Sampler en solo. Son jeu est fait d'un amas d'impulsions, espacées de 100 ms et mobiles en ambitus. Comme le montre la figure ci-dessous, le jeu est construit avec la répétition d'une impulsion grave accentuée.

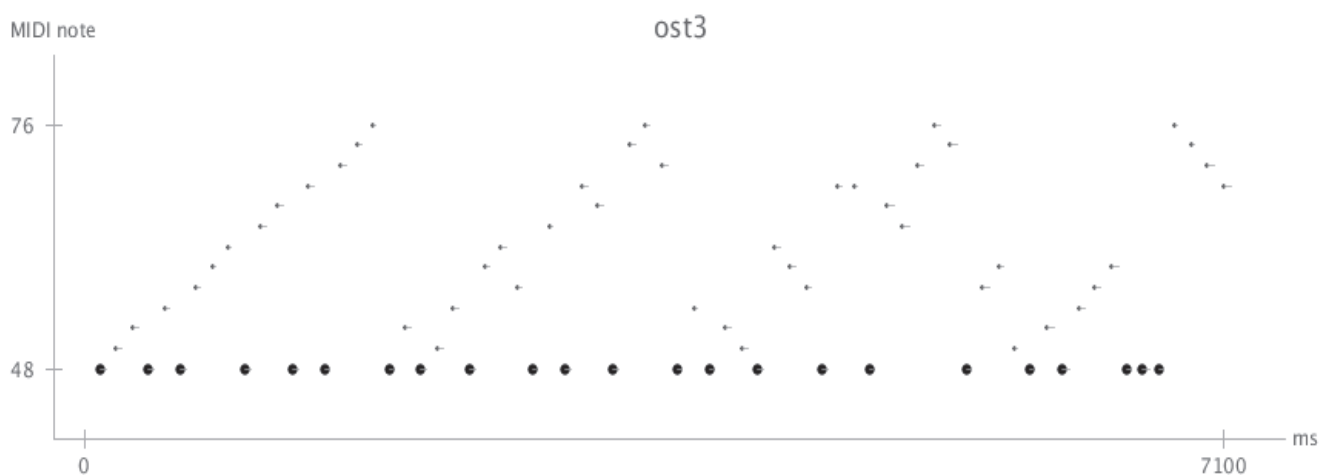


Fig. 85 - Ostinato du Sampler épisode IE-F

L'épisode (8D, evt 8-22) développe avec excès l'*ostinato* sur environ une minute. De la même façon que la section VII prolongeait la lumière d'une zone qui nous était familière, cet épisode prolonge (et assèche) cette zone de la perturbation. C'est donc le Sampler qui occupe la première place, avec un amas d'impulsions dont l'amplitude est maximale et avec un ambitus restreint.

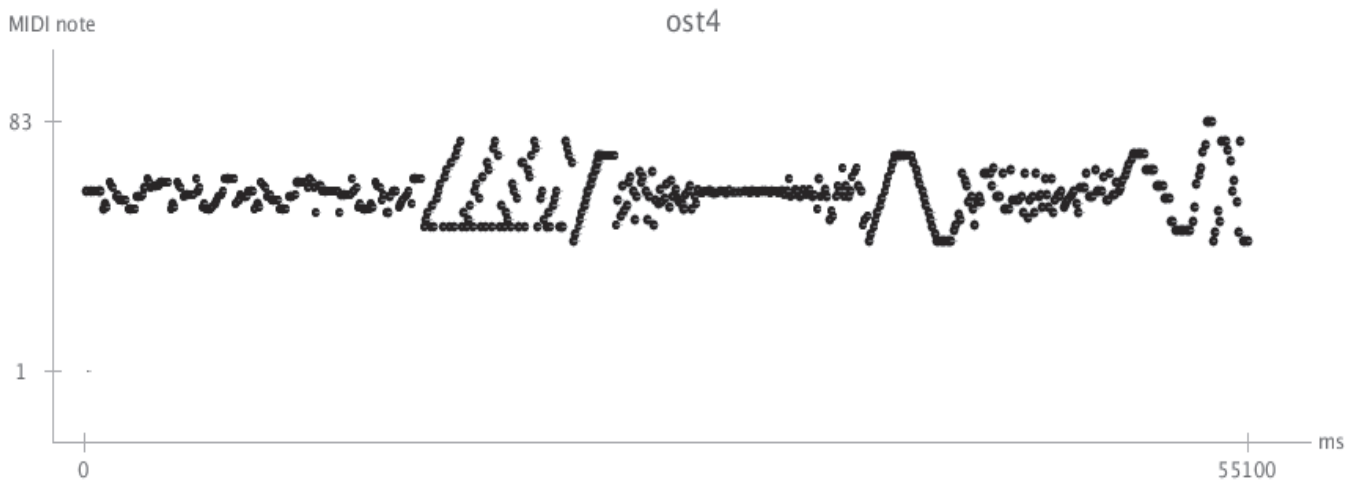


Fig. 86 - développement de l'Ostinato du Sampler épisode 8D

Dans ce développement la flûte n'a plus tellement sa place. À part quelques tenues *piano* elle joue régulièrement des croches très espacées, qui servent à déclencher des agrégats produits par le système TP. En parallèle de ces amas d'impulsions il y a donc le maintien d'agrégats comprimés. Contrairement aux agrégats aériens ceux-ci sont rythmiques, hachés, à une vitesse très rapide de sorte qu'ils fusionnent avec l'*ostinato*. Ceci est réalisé avec le Spat qui applique sur le système TP une spatialisation trop rapide²⁰⁹ pour être discernée : le résultat est ainsi similaire à un rapide trémolo qui hache un signal continu.

Après un tel développement l'épisode suivant (8E-F, evt 23-28) est dédié au déclin de l'*ostinato*. Il commence d'abord avec la flûte qui imite seule ce jeu en triples croches, puis par le Sampler qui lui répond mais qui va petit à petit s'essouffler. Le déclin de l'*ostinato* se fait avec un geste de l'aigu vers le grave, accompagné d'une irrégularité rythmique, comme le représente aisément la figure ci-dessous.

²⁰⁹ spatinc = 130

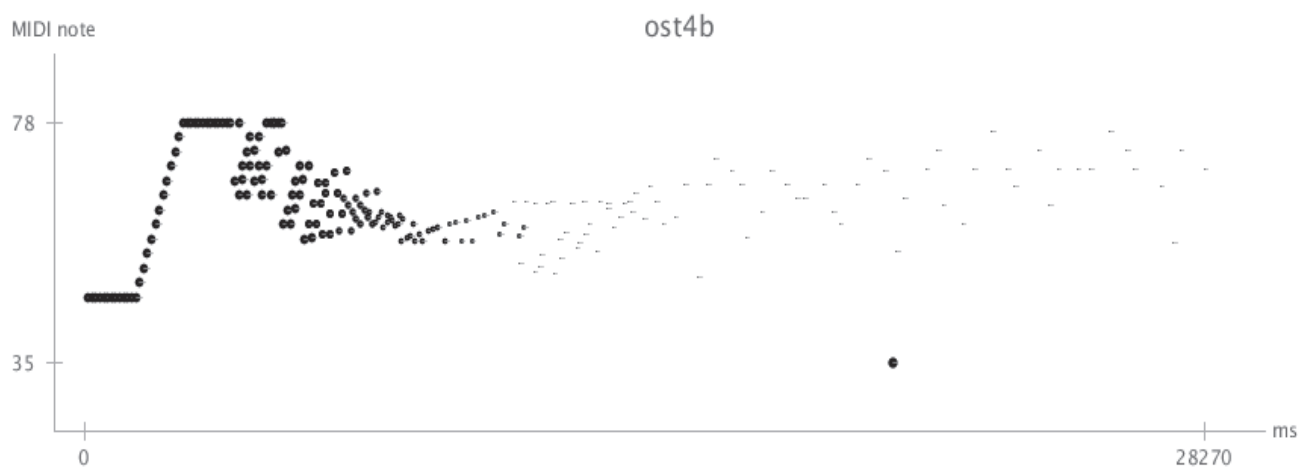


Fig. 87 - Déclin de l'Ostinato au Sampler, épisode 8E-F

La flûte rejoint le Sampler sur les dernières impulsions, en suivant ce geste de déclin, de sorte à mettre fin ensemble à cet *ostinato* qui a occupé toute la section.

Section IX

La section IX est une transition. Elle poursuit cette sensation de déclin présent à la fin de la dernière section mais d'une autre façon, avec les interpolations du Sampler (sample de piano). Elle fait ainsi écho à la V qui présentait les premières interpolations. Nous nous retrouvons ici à une frontière entre la stabilité et la perturbation, avec un matériau connu, celui des gestes proéminents du Sampler. Deux épisodes s'enchaînent, le premier poursuit le déclin et le second, par contraste, amène une perturbation.

À l'image du précédent épisode, la section commence avec un important geste de déclin, de l'aigu vers le grave, réalisé par des interpolations au Sampler. Ce geste néanmoins n'appelle pas l'arrêt, comme précédemment, mais il renouvelle un matériau que nous avons déjà rencontré à l'épisode 5A. Les deux premières interpolations de cet épisode (9A-B, evt I-II) font écho aux deux premières de l'épisode 5 où « quelque chose prenait forme ». Les gestes proéminents sont faits de pléthore d'impulsions, que l'on ne peut distinguer séparément, et qui diminuent petit à petit, comme en témoigne la représentation :



Fig. 88 - Geste du Sampler, épisode 9A

De même que dans l'épisode 5B, l'Harmonizer est utilisé pour épaissir ce nuage. Il grossit²¹⁰ sur toutes ses voies les impulsions. La flûte rejoint ce déclin sur la fin avec un matériau plus habituel, une trille sur la cellule c. L'épisode se poursuit avec une seconde interpolation qui commence d'abord par un élan, une montée linéaire, rapide, qui occupe tout le spectre, une sorte de relance, suivie immédiatement de nouveau d'un important déclin, de l'aigu vers le grave, avec une diminution du nombre d'impulsions.



Fig. 89 - Geste de déclin au Sampler, épisode 5B

Parallèlement la flûte rejoint ce déclin avec une succession de notes en point d'orgue, sans rythme et dans un large ambitus. Certaines de ces notes sont utilisées pour venir brouiller le

²¹⁰ Avec un retard de l'ordre de 100 ms et une transposition inférieur à une seconde majeure.

geste du Sampler avec le FS. Ce jeu de la flûte et ces brouillages permettent de nuancer le déclin, il l'éloigne du premier plan pour nous faire diriger vers un ailleurs et non vers une fin.

Le dernier épisode de la section (9C, evt 12-14) rompt ce déclin avec une succession de gestes très amples, comme nous avons pu en avoir dans l'épisode 5B.



Fig. 90 - Geste du Sampler, épisode 9C

Ces gestes ascendants et descendants sont donc très limpides. Ils sont plus rapides et surtout linéaires; ils ne correspondent plus au principe de « neumes » du compositeur. Ces mouvements viennent relancer la zone de la perturbation avec un rendu très machinal. Le dernier geste en *diminuendo* rejoint néanmoins le déclin global de la section.

La section se termine par une *coda* qui prolonge cet aspect machinal avec un *ostinato* à la flûte sur la note polaire Do#3, à laquelle se juxtapose la dernière interpolation, discrète, sans variation d'ambitus et située dans le grave. Elle appelle en fait la prochaine section qui reprend l'*ostinato*.

Section X

Cette section retourne dans la zone des perturbations avec soit des jeux en *ostinato* soit des gestes pléthoriques. C'est aussi une section de transition, elle réutilise des matériaux déjà rencontrés. Elle est faite de deux épisodes, le premier relance l'*ostinato* et le second ré-expose les gestes pléthoriques.

L'épisode (10A-B, evt 1-11) relance une dernière perturbation *d'ostinato* mais avec moins d'énergie que précédemment. Comme au début de la section VIII, l'épisode est fait d'un jeu de questions/réponses entre flûte et Sampler, qui se répondent par une série de triples croches.

Le rendu est cependant moins percussif puisque le Sampler active le sample de flûte basse, et d'autre part puisque son jeu est concentré dans le bas du spectre, ce qui le rend plus discret. La flûte occupe le premier plan avec la création d'un double par l'Harmonizer. Ce dernier en effet transpose à quatre reprises²¹¹ la flûte de sorte à lui créer un double grave. Enfin le Sampler et la flûte se rejoignent pour former un climax, venant marquer la fin de l'épisode. La flûte joue des doubles croches de plus en plus aiguës et surtout le Sampler effectue une montée linéaire. Cette montée, déjà très saillante, est augmentée, saturée, par la Reverb qui prolonge les impulsions du Sampler.

Le second épisode (10C-D, evt 12-15) réutilise les gestes linéaires, très amples en ambitus, exposés dans l'épisode 9C. Il y a d'abord une première montée, suivie d'une légère réponse à la flûte, puis une dernière longue phrase qui concrétise la perturbation. Cette phrase peut être vue comme un condensé de l'épisode 9C, comme en témoigne sa représentation.

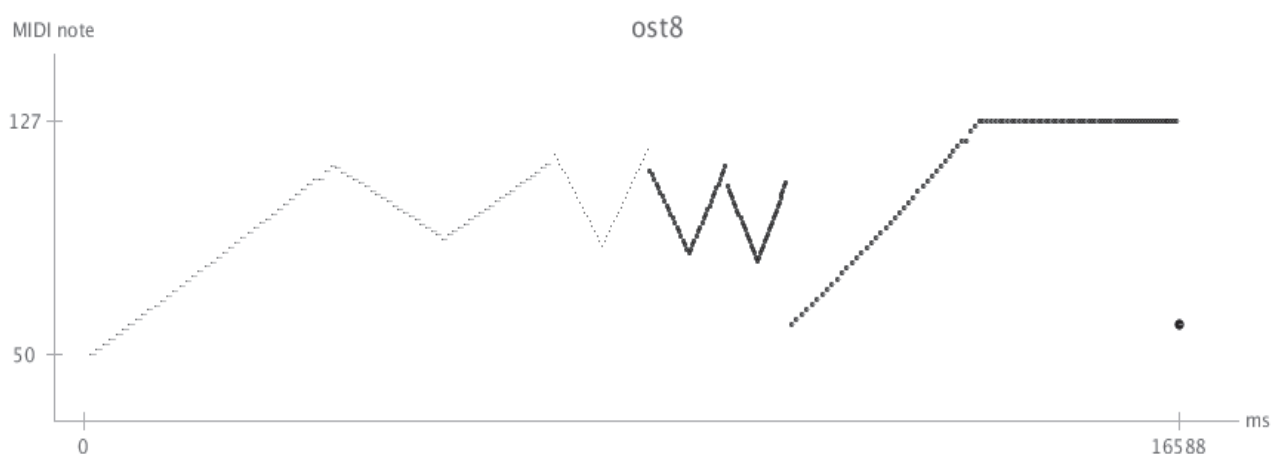


Fig. 91 - Geste du Sampler, épisode 10C-D

Section XI

La section XI quitte la zone des perturbations pour retrouver celle de la stabilité. On peut la comprendre comme une poursuite de la section IX, une poursuite du déclin de l'*ostinato*. Nous retrouvons maintenant le matériau connu de l'archétype *aérien*, celui des tapis aigus, « proche

²¹¹ -30, -16, -16 et -12 demi tons

du timbre de la flûte ». Cette section fait ainsi écho à la IV et à la VII avec cependant plus de perturbations. Elle est faite de quatre phrases qui respectent la même organisation.

La section commence à vive allure avec des triples croches à la flûte, qui provoquent, à l'aide du système TP, le premier tapis. Cette vivacité s'arrête net avec un silence marqué d'un point d'orgue, ce qui permet au tapis, augmenté de battements, d'occuper toute la place. Dans un second temps, sur le maintien du tapis, la flûte déroule un jeu plus vif, fait de triples croches en *accelerando* et *rallentando*, qui rappelle de loin l'*ostinato*.

Fig. 92 - Deuxième phrase de la section XI

Au fil de l'écoulement de ces phrases l'impression de stabilité est de plus en plus présente. Les tapis sont déclenchés de façon moins vive et la flûte passe des triples, aux doubles, puis des doubles aux croches et ainsi de suite jusqu'aux noires *flatterzunge*, typiques de l'archétype *aérien*. Nous retrouvons ici la même impression de stabilité que nous avons connue dans les précédentes sections, le temps de l'*ostinato* est révolu.

Section XII

Cette section nous amène pleinement dans la zone de la stabilité avec un ensemble de matériaux que l'on a déjà rencontrés dans les six premières sections. C'est en effet une des sections les plus riches en matériaux. Elle combine l'archétype *interaction continue* avec celui de la *continuité*. Les 15 phrases de la section ne sont ainsi pas construites selon un modèle, comme dans la VI, mais rappellent bien plus les trois premières sections. Nous retrouvons par exemple, en plus du Chapo et son enveloppe spectrale, les triples croches de l'Harmonizer. La section se compose de trois épisodes. Le premier expose le matériau, le deuxième amène une perturbation et le dernier développe cette perturbation.

Le premier épisode (12A-D, evt 1-4) expose le matériau de la continuité. Il commence lentement avec une consonance sur La₂ entre la flûte et le Chapo. L'exposition se poursuit avec un jeu de questions/réponses entre flûte et Chapo : des notes piquées de la flûte, construites autour de la cellule c (Mi₄, Sib₄) appellent des déplacements de l'enveloppe spectrale. L'épisode se termine avec un tempo plus soutenu, qui annonce la prochaine perturbation.

L'épisode (12E-H, evt 6-13) vient perturber la stabilité avec le matériau connu des triples croches de l'Harmonizer. La flûte lance l'Harmonizer avec des triples ascendantes et ce dernier y répond de même avec des triples sur-aiguës.

The image shows a musical score for episode 12E-H. It consists of two staves: the top staff is for the flute and the bottom staff is for the Chappo. The flute part has several measures with triplets, some marked with '(Harm.)' and 'long rub'. There are circled numbers 5, 6, and 7 under the flute staff. The Chappo part has triplets and dynamic markings like 'pp', 'mf', and 'p'. The score is labeled 'XII E' in a box at the top left.

Fig. 93 - extrait de l'épisode 12E-H

Comme le montre la figure, ce jeu est très présent puisque les triples de l'Harmonizer sont prolongées par la réverbération infinie. Le Chapo, de son côté, rejoint la perturbation avec des déplacements instantanés de l'enveloppe spectrale. Cette perturbation se poursuit avec un jeu *fortissimo* à la flûte. À l'événement II, cette dernière enchaîne 4 doubles stridentes, augmentées par des battements sur le Chapo, ce qui vient former un *climax*. La suite présente la chute de la perturbation : la flûte joue une longue phrase de triples croches, comme on en trouve dans l'archétype *aérien*, qui ouvre sur une phrase de « pause », faite uniquement de trois noires *pianissimo* où l'on peut lire « très lent et espacé » puis « sans expression, immobile » (p. 33)

L'épisode (12I-N, evt 14-19) poursuit la dernière perturbation. Il est essentiellement constitué d'un nouveau matériau, présenté dès le début, lent et menaçant, très bruité et situé dans le grave. Comme cet effet est unique, non réutilisé, il nous rappelle les *vaguelettes* de la section III, mais ici avec un rendu qui se situe à la limite de la stabilité. Cet effet est obtenu en appliquant tous les modules de traitement sur le Chapo avec le routing de la figure ci-dessous.

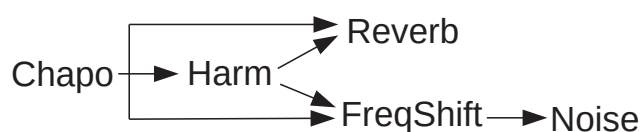


Fig. 94: Routing de l'épisode 12I-N

L'Harmonizer est utilisé pour transposer le spectre du Chapo – qui contient déjà les fréquences les plus basses de la section (cf. Chapitre VII 1.3.5.b) – à environ -5 et -8 demi tons. Ensuite, ce nouveau spectre est enrichi (i) par le FS qui ajoute des battements assez lents (5 Hz), et (ii) le Noise, qui brouille une énième fois le signal avec une valeur haute de *fnois*²¹². Enfin, la Reverb vient parallèlement enrichir le signal du Chapo en activant uniquement sa série de 6 filtres en peignes – *revfb* est en effet nul.

Fig. 95 - extrait bruité épisode 12I-N

Le rendu bruité de la phrase est annoncé à la flûte avec un *flutterzunge fortissimo* qui se fond dans le spectre bruité du Chapo. Ensuite la flûte enchaîne des notes très espacées en ambitus, de sorte à venir interagir avec le bloc sonore bruité. Ce rendu s'éteint à la phrase suivante et laisse place au matériau habituel qui va prolonger la perturbation. Celle-ci atteint son apothéose à la phrase M : la flûte enchaîne des triples croches *fortissimo* et surtout l'Harmonizer les prolonge avec 4 retards d'environ 350 ms.

La section se termine avec une *coda* qui nous replace pleinement dans la stabilité, en reprenant le jeu de questions/réponses qu'il y avait dans la *coda* de la section VI. Le spectre du Chapo est harmonique, comme à l'ouverture, bâti sur La₂, et la flûte joue des croches « très espacées et lentes » (p. 34). Le Chapo va suivre ces notes avec une synchronisation instantanée,

212 *Fnois* passe de 19 à 34 cf. Chapitre VII 2.4

tel un double électronique austère. En parallèle de ce jeu, des triples croches de l'Harmonizer sont ajoutées sporadiquement, prolongées par la Reverb elles viennent donner un rendu aérien.

Section XIII

La pièce se termine dans la stabilité d'une façon très épurée. La dernière section utilise en effet l'archétype *aérien*, avec un tempo plus lent (noire = 56) et de plus sans les motifs de triples croches usuels.

La section s'ouvre avec un La₄ à la flûte qui forme un tapis à l'aide de la Reverb infinie et le Paf. Sur ce tapis, à la fois aigu et épuré, la flûte déroule un jeu calme fait de croches. Deux notes de ces jeux sont utilisées pour alimenter le tapis. La seconde phrase respecte le même schéma, elle ajoute au début un soupçon de tension avec quelques triples croches qui seront éteintes dans trois noires *flutterzunge*. La pièce se termine avec un geste lent et fluide qui mène à un tapis aigu. La flûte joue la cellule c, *pianissimo*, puis l'Harmonizer poursuit avec un agrégat arpégé qui se prolonge dans la Reverb infinie. Sur ce dernier agrégat, très aigu, le FS ajoute des légers battements et le Spat, surtout, ajoute un lent déplacement dans l'espace stéréophonique.

The image shows a musical score for Section XIII C. It consists of two staves. The top staff is a flute line with notes marked 24, 25, and 26. Above the staff, there are handwritten annotations: '(Fl. → Rev. ∞)' above note 24, 'de plus en plus' above note 25, and '(Fl. → Harm.)' above note 26. To the right of the staff, there are more annotations: 'rvb-FS' in green, '(Harm. → Rev. ∞ → Freq. sh. 12 Hz)' in purple, and 'MIXER: très lent' with a slider icon. The bottom staff shows a piano line with dynamics *pp*, *pp* < *mf* > *pp*, and *mp*. A date stamp 'Paris le 27 mars 1987' is located at the bottom right of the score.

Fig. 96 - Phrase finale

La pièce se termine avec ce tapis aigu, aérien, qui s'efface petit à petit avec un long fondu de sortie. L'impression de sérénité est à son comble.

3 Vue globale

3.1 Sections et archétypes

Nous pouvons apporter une première vue globale de la pièce en représentant les liens que nous avons effectués entre sections et archétypes. C'est ce que propose la figure suivante, non en classant les sections mais en les situant dans un espace formé par les cinq archétypes.

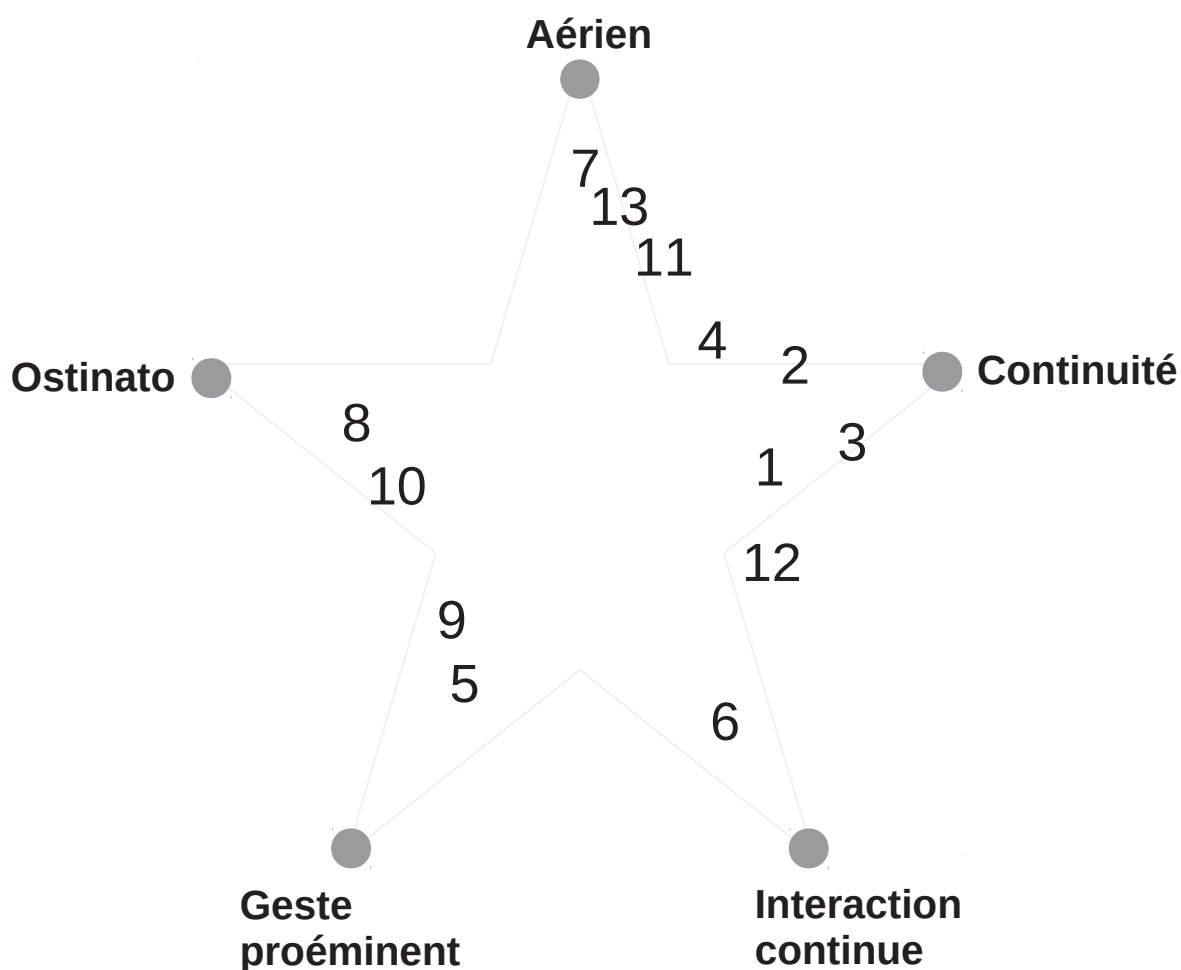


Fig. 97 - Placement des sections au sein de l'espace archétypal

Les sections correspondant le plus à un archétype, comme la VI et la VII, sont positionnées au plus proche de leur point, et, *a contrario*, les sections les plus riches se situent entre les archétypes, comme la XII.

3.2 Une structure ?

En reliant les 13 sections mentalement selon leurs successions nous pouvons identifier trois groupes. À droite de l'espace sont présentes les quatre premières sections ; sur la gauche les sections V à X – avec la VII qui est à l'extrémité gauche des sections se trouvant sur la droite ; puis, légèrement à gauche des quatre premières, les dernières sections XI à XIII.

Une telle répartition est par exemple présente dans l'analyse de Márta Grabócz²¹³. En poursuivant cette lignée l'auteur propose une forme Sonate pour la macrostructure de la pièce : exposition (I à IV), développement (IV à X), puis réexposition (XI à XIII). Il faut cependant noter que cette forme n'est pas proposée comme un plan formel, rigoureux, mais bien plus comme quelque chose qui se dessine avec des traits moins nets que d'autres :

[la macrostructure se présente à nous comme une] gigantesque forme sonate dont l'exposition, le développement et la reprise sont réalisés en plusieurs étapes intérieures et en plusieurs dimensions, comme si l'œuvre montrait « l'hologramme » d'un itinéraire de type « sonate » par rapport à ce que cette même forme représentait au cours des XIXe et XXe siècle. *Ibid.* p.94

Il y a bien en effet entre les sections VIII, IX et X quelque chose de l'ordre d'une réexposition, avec un fort poids structurel, mais la compréhension de ces 13 sections comme une forme Sonate pose aussi quelques problèmes, d'où cette idée d'« hologramme ». Les problèmes majeurs qui se posent découlent, selon nous, de passages qui ne sont pas réexposés. Ainsi de l'épisode 3F avec les vaguelettes qui n'apparaît nul par ailleurs. De même pour les gestes proéminents du Sampler qui, s'ils se répondent entre eux, ne sont pas exposés dans une première partie. Enfin, il en va de même pour la section VI, la plus structurée de toutes, où nous avons pu induire un certain modèle, dont on ne trouve pas de trace dans une première partie.

3.3 Un maillage de réminiscences

Afin d'éclairer la structuration de la pièce nous proposons de considérer un ensemble de réminiscences. Ces dernières n'ont pas uniquement lieu à l'échelle des sections mais aussi à celle des épisodes. Il y a en effet le long de ces 37 minutes un certain nombre de réminiscences, que nous avons identifiées dans l'analyse avec le terme *écho*. Dans la précédente figure de l'espace archétypal, ces réminiscences se retrouvent dans la proximité des sections.

Il est possible d'aller plus loin dans la représentation de ces réminiscences en considérant l'espace compositionnel que nous avons décrit en première partie. Il nous est alors nécessaire de

²¹³ [Grabócz 2013]. Il s'agit d'une monographie éditée en 2013, mais le chapitre qui concerne l'analyse de *Jupiter* a été réalisé dans les années 1990 (cf. nota p. 66). La méthode de synthèse PAF n'est ainsi pas considérée.

descendre dans la segmentation et de considérer maintenant les épisodes. Aussi nous constatons que ces réminiscences ne se dessinent pas seulement avec les matériaux, et/ou les archétypes, mais aussi dans le jeu de contraste entre *stabilité* et *perturbations*. La figure ci-dessous donne une représentation de ces réminiscences. Tous les épisodes sont situés dans cet espace selon la dimension horizontale (*stabilité* ~ *perturbations*) et les réminiscences entre épisodes sont indiquées par des courbes de Bézier.

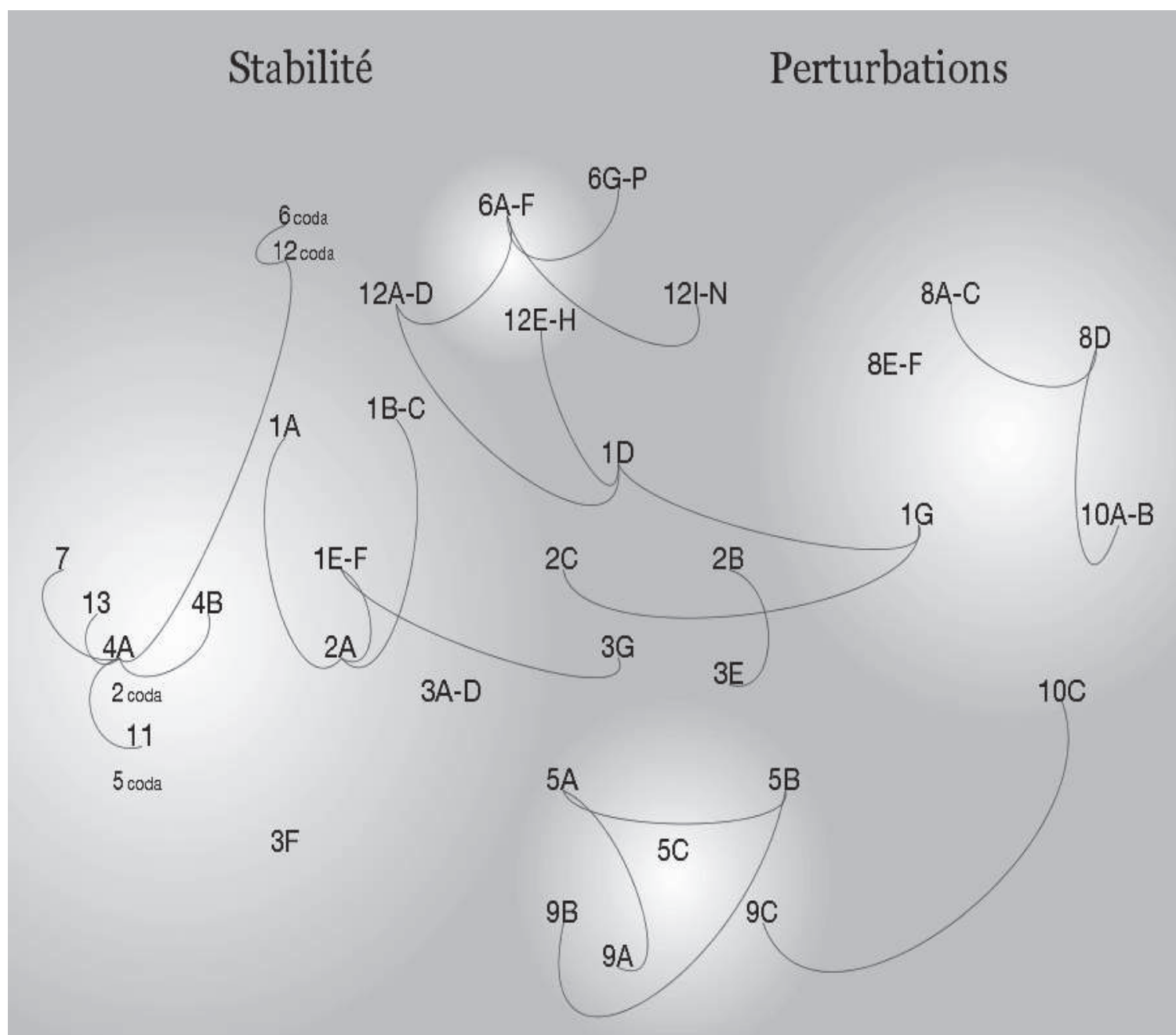


Fig. 98: Les réminiscences dans l'espace compositionnel

Nous comprenons alors par la redondance de certaines réminiscences que des structures musicales sont amenées à se dessiner. Ainsi [May, 2006, p. 182] voit dans les réminiscences présentes entre les épisodes *aériens* – représentés autour de 4A dans la précédente figure – les bribes d’une forme ritournelle. En contrepartie, par la disparité de ces épisodes, par l’isolement de certains (e.g. 3A-B, 3F, 5C), nous comprenons qu’une structure cohérente à l’échelle de la pièce reste difficilement applicable.

3.4 De l’expérimentation

Nous proposons d’expliquer ce maillage par une pratique compositionnelle qui se fait, notamment, par expérimentation. Cette expérimentation n’est évidemment pas le moteur premier de la composition, mais, à l’inverse, elle est présente au sein d’un contexte musical très profond, organisé, issu de la théorie musicale.

C’est peut-être dans cette expérimentation que *Jupiter* trouve une de ces originalités. Du fait de ce maillage entre phrases, épisodes et sections, on retrouve inévitablement, durant ces 37 minutes, des impressions de structures qui nous sont familières, mais, confrontées à un nouveau matériau dans les dizaines de minutes suivantes, ces structures s’effacent. Cette expérimentation, qui va résolument de pair avec une période pionnière donc, se retrouve dans des écrits du compositeur concomitants au cycle *Sonus ex Machina* :

Dans ce domaine [instrumental] la *mémoire* culturelle catalyse notre imagination car nous travaillons à partir d’objets connus et répertoriés. Ce n’est absolument pas le cas avec les nouveaux matériaux. Nous n’avons, dans ce cas, aucune histoire, ou une histoire tellement récente et éclatée, qu’elle peut difficilement agir sur notre comportement. Certes il existe des méthodes de synthèse très diverses et très spécifiques ayant chacune leur propre mode de fonctionnement et leurs propres contrôles, mais le matériau sonore lui-même étant tellement spécifique à une œuvre donnée qu’il me paraît impensable de le réutiliser dans un autre contexte sans transformations substantielles. De ce point de vue, la meilleure méthode d’investigation, la mémoire faisant défaut, reste l’*expérimentation* [Manoury, 1998, p. 55] (souligné dans le texte).

S’il ne m’est pas donné de dire en quoi cette esthétique peut consister, j’aimerais juste relever qu’il s’agit pour moi d’une *problématique de l’instant*. Cela signifie que, même à mon corps défendant, je suis amené à travailler de plus en plus localement sur des propositions qui se font jour dans le cadre d’une expérimentation. *Ibid.* p. 57 (souligné dans le texte)

Conclusion

Notre problématique était construite sur une prémisse, celle que l'artefact peut être utile pour analyser la musique d'informatique. À partir de là nous nous sommes demandé « comment cet artefact peut-il être utilisé pour aider l'analyse ? »

Notre première partie était une approche de l'artefact ; un ensemble de premières réflexions qui nous ont permis d'avoir un aperçu global. Dans un premier temps nous avons esquissé une grille montrant que les artefacts (i) peuvent avoir plusieurs apparences (ii) cristallisent des connaissances, (iii) sont le fruit d'un travail de design, (iv) intègrent une sémantique et (v) sont d'une importante densité. Dans un second temps l'artefact a été investi à l'aide de travaux existants. Il nous a fallu de prime abord éloigner l'artefact d'une étude trop focalisée et à l'inverse l'intégrer au sein d'un système, le système HAS, avec lequel il prend sens. Nous avons ensuite utilisé les travaux de [Vinet, 2003], [Orlarey, 2006] et le célèbre [Winograd *et al.*, 1996] pour éclairer une structuration en niveaux d'abstraction : l'artefact n'est pas quelque chose de plat, sur lequel on projette des idées musicales, mais c'est un medium qui possède une profondeur, et la projection des idées musicales passe par l'investissement de cette profondeur.

Dans la deuxième partie nous avons répondu à notre problématique avec des propositions théoriques. La branche Homme/Son du système HAS a été investie à l'aide de la notion de matériau musical. Cette notion – qui se dessine elle-même avec une profondeur – nous a permis de faire correspondre à l'acte compositionnel un processus hautement dynamique. La composition, l'élaboration du matériau, se réalise avec un processus dynamique, entre idées compositionnelles, artefacts et rendu sonore. Ce processus a ensuite été investi avec la sémiotique du philosophe Charles Sanders Peirce. Ceci nous a permis de distinguer trois moments dans l'acte compositionnel, qui sont, réciproquement, trois étapes que l'analyste doit franchir pour saisir l'artefact. Ces trois étapes se dessinent dans la succession de trois termes clés en informatique musicale : *variable*, *paramètre* et *contrôle*. Alors nous avons vu que le travail de l'analyste doit tendre vers la mise au clair des *contrôles* de l'artefact. Cependant, même doté d'une direction le travail, l'exercice d'analyse n'a rien d'évident face à la densité de l'artefact.

Afin de guider cette mise au clair des contrôles nous avons défini un cadre conceptuel qui puisse permettre, de façon cohérente, de décrire les artefacts des pièces musicales. L'utilité d'un tel cadre se trouve essentiellement dans le premier point de singularité que nous avons relevé, à savoir les artefacts possèdent des apparences ; et de ce constat découle le suivant : la structuration de leur apparence ne reflète pas nécessairement leur structuration. La description à effectuer doit par conséquent dépasser l'implémentation à laquelle l'analyste est confronté, de sorte à atteindre les idées musicales véhiculées par l'artefact. Le cadre que nous avons proposé est constitué de trois éléments, des Modules, des Algorithmes et un Routing. Ces éléments ne sont pas à identifier dans l'implémentation mais c'est à l'analyste de les construire lors de son travail : à l'instar du travail compositionnel, l'identification des Modules et des Algorithmes doit se faire au sein d'un processus hautement dynamique. Les Modules peuvent ainsi imbriquer des Algorithmes autant que nécessaire. Cet aspect dynamique du cadre étant tout aussi primordial que difficile à cerner, nous l'avons éclairé avec le couple de concept potentiel/actualisation, présent chez M. Stroppa. Dans la composition les Modules sont attachés à des potentiels, un ensemble de sons en puissance qui partagent, lors de leur actualisation, des propriétés communes. Nous avons ensuite enrichi ce couple de concepts – entre quelque chose d'abstrait et quelque chose qui se perçoit – par deux couples de concepts présents dans l'étude du langage, la Langue et la Parole (Saussure) et la Compétence et la Performance (Chomsky). Ces deux paires de concepts nous ont permis de finaliser le premier élément de réponse à notre problématique. L'analyste doit dans un premier temps rendre compte de la structure de contrôle de l'artefact, c'est-à-dire à la fois l'ensemble des contrôles sur lesquels le compositeur a exercé une action, et l'usage qui a été fait de cet ensemble de contrôles. Le couple Chomskyen Compétence/Performance nous a en effet amenés à considérer l'usage qui a été fait des contrôles. Ces derniers sont signifiants non pas en eux-mêmes, mais bien dans l'usage qu'en a fait le compositeur, c'est-à-dire au fond dans ce que ces contrôles représentaient lors de la composition.

Nous avons ensuite investi le champ de l'analyse musicale afin de formuler le second élément de réponse de notre problématique. Un premier schéma partial nous a permis de différencier deux pratiques dans l'analyse des musiques électroacoustiques, schématiquement des analyses basées sur ce qu'il y a en amont du signal audio, et d'autres basées sur les signaux audios eux-mêmes – que cela soit en analysant ces signaux (e.g. les descripteurs audio) ou bien en les utilisant pour naviguer dans l'écoute (e.g. analyse aurale). Ce schéma est partial puisqu'il se base sur le principe que l'analyse des musiques électroacoustiques s'est développée à partir d'une absence de partition graphique. Nous avons ensuite pointé une spécificité primordiale de la musique d'informatique : elle est écrite. Une écriture qui se fait non pas au sein de la page, ou avec des inscriptions graphiques, mais une écriture alphanumérique, ou computationnelle, avec laquelle la composition se fait. Le second élément de réponse à notre problématique se trouve à

l'intersection du précédent schéma partial. Après un travail fortement ancré sur l'artefact, sur des « signes technologiques » pourrait-on dire, l'analyse doit se faire dans l'écoute. Une fois que la structure de contrôle a été mise au clair, l'analyse doit la confronter au sonore de la pièce, elle doit ainsi répondre aux questions « comment cela a été fait ? », « comment cela a été construit ? » et ceci dans la durée de la pièce.

Dans la troisième partie nous avons appliqué nos propositions théoriques sur la pièce *Jupiter*. Dans un premier temps une introduction générale sur la pièce a été donnée. Il fallait en effet resituer cette pièce dans cette volonté *d'interaction* entre instrumentiste et électronique ; témoigner de son passé foisonnant de technologies et expliquer les supports que nous avons utilisés. Nous avons ensuite donné un aperçu global de l'artefact. Les différents Modules et les deux Algorithmes ont été abordés globalement et nous avons présenté le routing avec le problème amené par son implémentation. À la suite de cette introduction nous avons appliqué le premier élément de réponse à notre problématique : mettre à jour la structure de contrôles de l'artefact. Chacun des Modules a été décrit avec le cadre théorique défini au chapitre IV. En suivant l'aspect dynamique de ce cadre nous avons imbriqué les deux algorithmes aux deux Modules auxquels ils se rattachent. Ensuite nous avons donné une représentation de la structure de contrôle. Au cours de cette application nous nous sommes aperçus que les Modules pouvaient avoir des usages très différents si bien qu'il peut être nécessaire de les subdiviser. Ce fut par exemple limpide pour l'Harmonizer, qui permet tantôt de transposer à quatre reprises une note de flûte et tantôt de flouter un signal. Dans la même lignée nous nous sommes aperçus que les contrôles pouvaient avoir des « poids de significations » très différents ; tous n'ont en effet pas la même importance. La représentation de la structure de contrôle donnée tient ainsi compte de ces différents poids de signification. On retrouve en haut les Modules qui imbriquent des algorithmes, les plus saillants, puis à l'opposé en bas ceux qui relèvent du mixage.

Pour mener à bien l'analyse de la pièce nous avons eu besoin de réaliser trois programmes informatiques. Le premier était utile au début du travail pour s'approprier l'artefact et ses nombreuses variables. Le deuxième était utile pour l'analyse, il a permis de donner une écriture plus condensée du Paf, qui mobilise à lui seul une trentaine de variables. Enfin le troisième programme était spécifique à *Jupiter* puisqu'il a permis de faire face au problème du *routing*. Nous avons programmé une structuration par Module et appliqué quelques règles de cohérence de façon à mettre au clair les différentes chaînes de routing actives à un événement donné. Le fichier obtenu nous a ainsi permis de savoir quels Modules étaient actifs et avec quels embranchements. C'est avec un tel fichier que nous avons pu expliquer les sonorités perçues lors de l'analyse, ou encore rendre compte de la construction de certains passages.

Dernièrement, nous avons appliqué le second élément de réponse de notre problématique ; nous avons analysé la pièce avec les connaissances acquises lors de la mise au clair de la structure de contrôle. Notre analyse s'est faite en trois temps. Dans un premier nous avons décrits les éléments perceptifs de la pièce en se représentant un espace compositionnel. Cet espace contient un ensemble de *matériaux*, qui se caractérisent par leur sonorité, lesquels sont regroupés dans des *archétypes*, qui rendent compte des idées compositionnelles. Dans le deuxième temps nous avons analysé la pièce section par section. Nous avons expliqué ce qui est perçu par la production, par l'artefact, puis plus globalement nous avons témoigné des jeux d'opposition qui ont permis d'expliquer les enchaînements d'épisode. Dans le dernier temps la structure globale de la pièce a été éclairée. Plutôt que d'apposer une forme cohérente nous avons proposé un maillage de réminiscences qui a lieu sur la durée de la pièce. Nous avons alors expliqué ce dernier à l'aide d'une pratique compositionnelle qui se fait, au sein d'un contexte profond forgé par la théorie musicale, par expérimentation.

La voie de cette thèse a été de confronter deux domaines que l'on peut souvent rencontrer comme séparés, celui de la computation sonore/musicale, du calcul des signaux audio au sein d'un contexte musical, et celui de l'analyse musicale. Notre travail hérite ainsi des travaux pionniers de [Di Scipio, 1995] jusqu'à plus récemment [Clarke, 2012]. Il nous paraît néanmoins que cette conjonction, entre appropriation d'un artefact et analyse musicale de la pièce à laquelle il se rattache, reste peu courante. La compréhension de l'artefact permet en effet de rendre compte de tout un arsenal de technologies musicales, sans qu'il puisse sembler important, par la suite, de les confronter à la dimension canonique de la musique. Questionner la rencontre de ces deux « pratiques musicologiques » nous paraît fécond. La saisie de l'utilisation faite par le compositeur de tels « producteurs sonores », et la saisie entière des algorithmes implémentés, permettent d'atteindre une certaine profondeur musicale²¹⁴, chère à l'analyse comme l'avait remarqué [Risset, 2001]. Cette profondeur permet d'éclairer les représentations musicales avec lesquelles la composition s'est réalisée, et grâce au medium numérique – vu comme une objectivation de la pensée –, de façon non ambiguë. Il nous semble que la pratique de l'analyse avec de telles représentations peut être féconde pour l'établissement d'un répertoire des musiques d'informatique, et par là même pour la création musicale.

²¹⁴ Le démantèlement des algorithmes nous a permis d'être au plus proche des idées qui ont régi la composition, et par là même des prémisses d'une musique en temps réel.

Table des Figures

Figl. 1 - Schéma de synthèse pour simuler des sons de cloches, extrait de [Dodge, 1997].....	36
Fig. I2 - table de donnée pour simuler des sons de cloches, extrait de [Dodge, 1997].....	37
Fig.3 - Le système HAS.....	54
Fig. 4: Exemple de représentation pour la pièce Solitude (2007) de Hans-Christoph Steinner...	66
Fig. 5: trois natures d'information et leurs positionnements, d'après [Orlarey, 2009].....	72
Fig. 6 - Première page de Studie II (1954), de K. Stockhausen. en haut l'espace des fréquences, au milieu celui du temps, en bas celui de l'amplitude © Universal Edition (London).....	86
Fig. 7-Le système HAS, au sein duquel le matériau se conçoit.....	89
Fig. 8 - Exemple d'entité et de relation du modèle FRBR. capturé depuis [Leresche et al., 2012, p.14].....	105
Fig. 9 - Analogie entre le modèle FRBR et un cadre conceptuel pour décrire les artefacts.....	107
Fig. 10 - Représentation du cadre conceptuel au sein du système HAS.....	117
Fig. 11 - Relations entre performance, compétence et grammaire.....	130
Fig. 12 - Le couple de concepts potentiel et actualisation.....	133
Fig. 13 - Positionnement de la structure de contrôle au sein du cadre conceptuel.....	135
Fig. 14 - la machine 4X et la flûte 4X, vue du musée © Musée de la musique – Philharmonie de Paris.....	167
Fig. 15 - La disposition des Haut-Parleurs.....	171
Fig. 16 - Extrait de la partition manuscrite, Section 1.....	173
Fig. 17 - Le système de synchronisation.....	175
Fig. 18 - Le système de synchronisation et les Modules.....	179
Fig. 19 - Implémentation du routing interne dans le patch.....	180
Fig. 20 - Le routing externe.....	183
Fig. 21 - Comparaison de la durée et du nombre d'événements par section.....	186
Fig. 22 - Paramètres de l'enveloppe d'amplitude appliquée sur chacune des voies du Sampler	190

Fig. 23 - Vue en boîte de l'algorithme effectuant les interpolations.....	191
Fig. 24 - Carte de fréquences. Capturé de [Manoury et al., 2003, Glossaire].....	194
Fig. 25 - Représentation de la première interpolation (inter1a) dans la partition gravée.....	195
Fig. 26 - Représentations des séquences. En haut une séquence d'interpolation ; en bas une ostinato.....	197
Fig. 27 - Note polaire déclenchée par l'intermédiaire du patch. A gauche le patch Pure Data ; à droite la notation musicale extraite de la partition. section II, événement 77.....	198
Fig. 28 - Schéma de synthèse du Module Additive. Les variables présentes dans le qlist sont indiquées en italique.....	199
Fig. 29 - L'enveloppe spectrale de Chapo.....	202
Fig. 30 - schéma de synthèse de chapo. Les variables en italiques sont celles présentes dans les qlist.....	203
Fig. 31 - Les étapes de l'algorithme du Chapo. Les variables en italiques sont celles présentes dans les qlsit.....	204
Fig. 32 - relevé du facteur f porta = 1024 abscisse : temps (sec) ; ordonnée : facteur f noire : valeur de destination ; gris : facteur f.....	206
Fig. 33 - Les stack de fréquences de la section VI abscisse : événements ; ordonnée : fréquences.....	210
Fig. 34 - Description de formants, extrait de [Puckette, 2006, p. 148] f : fréquence, a : amplitude, b : largeur de bande.....	211
Fig. 35 - Schéma de synthèse général du Paf.....	213
Fig. 36 - Génération du signal porteur.....	214
Fig. 37 - Génération du signal modulant.....	215
Fig. 38 - Schéma de traitement d'une voie de l'Harmonizer.....	220
Fig. 39 - Schéma de traitement global du Noise.....	227
Fig. 40 - Représentation de deux déplacements. Un déplacement est fait de la combinaison de deux tables verticales. En haut les tables pour x ; en bas les tables pour y.....	230
Fig. 41 - Les significations présentes dans la structure de contrôle. Les voies des Modules et les contrôles d'amplitude sont ignorés.....	232
FigVIII. 42 - Entrée et Sortie du programme extractVarsFromQlist.....	237
FigVIII. 43 - Entrée et sorties du pogramme showModScore.....	239
Fig. 44 - Écritures de la Reverb sur la durée de la pièce. La durée est en abscisse sous la forme section.événement.....	241
Fig. 45 - Représentation des écritures du Noise sur la durée de la pièce. La durée est en abscisse sous la forme section.événement.....	242
Fig. 46 - Entrée et sortie du programme pafScore.....	243
Fig. 47 - Schéma UML, structuration du programme pafScore.pde.....	244
Fig. 48 - Entrée et sortie du programme preStateOfMods.....	248
Fig. 49 - Schéma UML de la structuration du programme PreStateOfMods.pde.....	249

Fig. 50 - la classe Module.....	252
Fig. 51 - États des Modules sur la durée de la pièce. Image découpée pour l'affichage.....	254
Fig. 52 - États des Modules sur la section II un point est un événement.....	255
Fig. 53 - Représentation du routing. Section VI événement 6 et 13 Les modules suivis d'un point noir sont reliés à la diffusion (dac).....	256
Fig. 54: Le système STR de la partition traditionnelle et le code couleur utilisé.....	258
Fig. 55 : Augmentation de la représentation du routing dans la partition traditionnelle.....	259
Fig. 56 : Représentation du Routing sur la partition traditionnelle depuis nos résultats.....	259
Fig. 57 - Phrase d'ouverture.....	264
Fig. 58-Installation de notes polaires. Sous section IA.....	264
Fig. 59 - Motif α (gauche) et β	265
Fig. 60 - Les triples de l'Harmonizer.....	266
Fig. 61 - Les deux moyens de production des agrégats, le système TP suivi du Paf.....	267
Fig. 62 - Les deux brouillages du FS.....	268
Fig. 63 - Exemple d'une phrase type de l'archétype aérien.....	269
Fig. 64 - Jeu de questions/réponses avec l'archétype Ostinato.....	270
Fig. 65-Épisode 1A.....	272
Fig. 66 - Épisode 1B-C.....	273
Fig. 67 - Épisode 1D.....	274
Fig. 68 - Épisode 1E-F.....	275
Fig. 69 - Épisode 1G. Les éclats du Paf.....	275
Fig. 70 - Épisode 2A.....	276
Fig. 71 - Épisode 2A.....	277
Fig. 72 - Épisode 2C.....	277
Fig. 73 - début de la deuxième phrase de 3A-D.....	279
Fig. 74 - Développement de l'épisode 3F.....	280
Fig. 75 - Climax de l'épisode 3F.....	281
Fig. 76 - Début de l'épisode 4A.....	282
Fig. 77 - Geste proéminent du Sampler, épisode 5A.....	283
Fig. 78 - Geste proéminent Sampler, épisode 5B.....	284
Fig. 79 - Forme d'onde et spectre du Do3 de la flûte Les lignes noires verticales représentent 50 ms Fenêtre de Hanning, 2048 échantillons.....	285
Fig. 80 - Phrase type du modèle tensions/détentes.....	285
Fig. 81 - Extrait de l'épisode 6A-F.....	287

Fig. 82 - Épisode 6G-P.....	288
Fig. 83 - Deuxième phrase de la section VII.....	289
Fig. 84 - extrait épisode 8A-C.....	290
Fig. 85 - Ostinato du Sampler épisode IE-F.....	290
Fig. 86 - développement de l'Ostinato du Sampler épisode 8D.....	291
Fig. 87 - Déclin de l'Ostinato au Sampler, épisode 8E-F.....	292
Fig. 88 - Geste du Sampler, épisode 9A.....	293
Fig. 89 - Geste de déclin au Sampler, épisode 5B.....	293
Fig. 90 - Geste du Sampler, épisode 9C.....	294
Fig. 91 - Geste du Sampler, épisode 10C-D.....	295
Fig. 92 - Deuxième phrase de la section XI.....	296
Fig. 93 - extrait de l'épisode 12E-H.....	297
Fig. 94: Routing de l'épisode 12I-N.....	298
Fig. 95 - extrait bruité épisode 12I-N.....	298
Fig. 96 - Phrase finale.....	299
Fig. 97 - Placement des sections au sein de l'espace archétypal.....	300
Fig. 98: Les réminiscences dans l'espace compositionnel.....	302

Bibliographie

ABELSON, Harold, SUSSMAN, Gerald Jay, SUSSMAN, Julie. 1984. *Structure and interpretation of computer programs*. Cambridge, MIT Press.

ASHBY, William Ross. 1957. *An introduction to cybernetics*. Londres, Chapman & Hall Ltd.

ASSAYAG, Gérard. 1993. "CAO : vers la partition potentielle". In *La Composition Assisté par Ordinateur*. Cahiers de l'Ircam, no 3. Paris, Ircam - Centre Georges Pompidou.

ASSAYAG, G. 2014. "Introduction". In *Technique et Science Informatiques*, vol. 33, no 7-8 (Informatique Musicale), p 563-566.

BARTHES, Roland. 1964. "Éléments de sémiologie". In *Communications*, vol. 4, no 1, p 91-135. DOI [10.3406/comm.1964.1029](https://doi.org/10.3406/comm.1964.1029)

BARRIÈRE, Jean-Baptiste. 1986. "Mutations du matériau, mutations de l'écriture". In *Inharmoniques*, no 1 (Le temps des mutations). Paris, Ircam - Centre Georges Pompidou.

BARRIÈRE, J.-B. 1989. "L'informatique musicale comme approche cognitive : simulation timbre et processus formels". In *La musique et les sciences cognitive*, dir. Steven McAdams et Irène Deliège. Bruxelles, Madraga.

BARRIÈRE, J.-B. 1993. "Écriture musicale et ordinateur, notes sur la genèse". In *Genesis*, no 4 (Manuscrits – recherche – invention), p 11-22.

BATESON, Gregory. 1977. *Vers une écologie de l'esprit*. Vol. I et II. Paris, Éditions du Seuil.

BATTIER, Marc. 2003. "A Constructivist approach to the analysis of electronic music and audio art-between instruments and faktura". In *Organised Sound*, vol. 8, no 3, p 249-255. DOI [10.1017/S1355771803000220](https://doi.org/10.1017/S1355771803000220)

BATTIER, M. 2015. "Describe, Transcribe, Notate : Prospects and problems facing electroacoustic music". In *Organised Sound*, vol. 20, no 1, p 60-67. DOI [10.1017/S1355771814000430](https://doi.org/10.1017/S1355771814000430)

BAUDOIN, Olivier. 2007. "A reconstruction of *Stria*". In *Computer Music Journal*, vol. 31, no 3, p 75-81. DOI [10.1162/comj.2007.31.3.75](https://doi.org/10.1162/comj.2007.31.3.75)

BAUDOIN, O. 2009. "La faktura, «outil conceptuel d'analyse» –illustration avec *Stria*, de John Chowning". In *Journées d'Informatique Musicale*, Grenoble.

BAUDRY, Olivier, BONARDI, Alain. 2015. "Le frequency shifter : enquête dans un repertoire d'œuvres mixtes produites à l'Ircam". In *Journées d'Informatique Musicale*, Montréal.

BAYLE, François. 1989. "L'image de son, ou i-son. Métaphore/Métaphorme". In *La musique et les sciences cognitives*, dir. S. McAdams et I. Deliège. Paris, Mardaga, p 235-242

BAYLE, F. 2006. "Une logique des sensations". In *Lien*, no 4 (L'analyse perceptive des musiques électroacoustiques), p 112-119.

BAZOLLI, François, FABRE, Thierry. 2000. "Jean-Claude Risset". In *La pensée de midi*, vol. 2, no 2, p 66-73.

BEAUGÉ, Pascal. 2002. "Un savoir Musical : la notion de hauteur". In *Journal de Recherche en Éducation Musicale*, vol. 1, no 1, p 55-84.

BERGSLAND, Andreas. 2011. "The Six Fantasies Machine – an instrument modelling phrases from Paul Lansky's *Six Fantasies*". In *New interfaces for musical expression*, Oslo.

BIRNBAUM, David, FIEBRINK, Rebecca, MALLOCH, Joseph, WANDERLEY, Marcelo M. 2005. "Towards a dimension space for musical devices". In *New interfaces for musical expression*, Singapore, p 192-195.

BONARDI, Alain, ROUSSEAU, Francis. 2011. "L'émergence de pratiques musicales a-musicologiques". In *Musiker Cuadernos de música*, no 18, p 99-115. [hal-01156706](https://doi.org/10.1156706)

BONARDI, A., DUFEU, Frédéric. 2013a. "How Can We Model Behaviours of Digital Instruments? Propositions in fuzzy logic from Philippe Manoury's works". In *Electroacoustic Music Studies*, Lisbonne.

BONARDI, A. 2013b. "Pérenniser pour transmettre, transmettre pour pérenniser - Destins de l'œuvre mixte interactive - Autour de *En Echo*, pièce de Philippe

Manoury”. In *Musique et technologie - Préserver, archiver, reproduire*, Portraits Polychromes, hors-série thématique, Paris, INA-GRM, p 105-126.

BONARDI, A., DUFEU, F. 2015. “Rejouer une oeuvre avec électronique temps réel : enjeux informatiques et musicologiques”. In *Genèses musicales*, dir. N. Donin, A. Grésillon, J.-L. Lebrave. Paris, Presses Universitaires de Paris-Sorbonne, p 239-254. [hal-01242545](#)

BONARDI, A. 2017. “Analyser l'orchestre électronique interactif dans les œuvres de Manoury”. In *Analyser la musique mixte*, Sampzon, Delatour. [hal-01575296](#)

BRESSON, Jean. 2007. *Sound synthesis in computer-aided composition*. PhD, Université Pierre et Marie Curie.

BRESSON, J., AGON, C. 2009. “Synthèse sonore et composition musicale : problématique et outils d'écriture électroacoustique”. In *Electroacoustic Music Studies*, Buenos Aires.

BUDÓN, Osvaldo. 2007. “Composer avec des objets, réseaux et échelles temporelles : une interview avec Horacio Vaggione”. In *Espaces composables. Essais sur la musique et sur la pensée musicale d'Horacio Vaggione*, dir. Makis Solomos. Paris, L'Harmattan, p 101-119.

CHION, Michel. 1983. *Guide des Objets Sonores*. Paris, Editions Buchet.

CHION, M. 1991. *L'art des sons fixés, ou, La musique concrètement*. Rives, Metamkine.

CHOMSKY, Noam. 1957. *Syntactic structures*. La Haye, Mouton. Traduit par M. Braudeau, *Structures syntaxiques*. Paris, Le Seuil, 1969.

CHOMSKY, N. 1964. *Current Issues in Linguistic Theory*. La Haye, Mouton.

CHOMSKY, N. 1965. *Aspects of the theory of syntax*. Cambridge, MIT Press. Traduit par J.-C. Milner, *Aspects de la théorie syntaxique*. Paris, Le Seuil, 1971.

CHOMSKY, N. 1967. “The formal nature of language”. In *Biological foundations of language*. New York, John Wiley and sons. Inc. Traduit par D. Nelcya et S. Dan, *La linguistique cartésienne. Un chapitre de l'histoire de la pensée rationaliste, suivi de La nature formelle du langage*. Paris, Le Seuil.

CHOMSKY, N. 1996. “Language and Thought : Some Reflections on Venerable Themes”. In *Powers and Prospects : Reflections on Nature and the Social Order*. London, Pluto Press.

CLARK, Lynn. 2007. “Cognitive sociolinguistics : A viable approach to variation in linguistic theory”. In *LACUS Forum*, no 33, Houston TX.

CLARKE, Michael. 2005. "An interactive aural approach to the analysis of computer music". In *International Computer Music Conference*, Barcelona.

CLARKE, M. 2006. "Jonhatan Harvey' s *Mortuos Plango, Vivo Voco*". In *Analytical methods of electroacoustic music*, dir. M. Simoni. Abingdon, Routledge.

CLARKE, M. 2009. "Extending Interactive Aural Analysis : Acousmatic Music. In *Electroacoustic Music Studies Network*, Buenos Aires, Argentina.

CLARKE, M. 2010. "Wind Chimes, and interactive aural analysis". In *Denis Smalley, Portrait Polychromes*, Paris, INA-GRM.

CLARKE, M. 2012. "Analysing Electroacoustic Music : an Interactive Aural Approach". In *Music Analysis*, vol. 31, num 3, p 347-380. DOI [10.1111/j.1468-2249.2012.00339.x](https://doi.org/10.1111/j.1468-2249.2012.00339.x)

COGAN, Robert. 1985. *New Images of musical sounds*. Cambridge, Harbard University Press.

COLLINS, Nick. 2008. "The analysis of generative music programs". In *Organised Sound*, vol. 13, no 3, p 237-248. DOI [10.1017/S1355771808000332](https://doi.org/10.1017/S1355771808000332)

COLLINS, N. 2010. *Introduction to computer music*. Hoboken, John Wiley & Sons.

CONT, A. 2008. *Modeling Musical Anticipation : From the time of music to the music of time*. PhD, Université Paris 6, Paris.

COUPRIE, Pierre. 2003. "Analyse comparée des *Trois rêves d'oiseau* de François Bayle". In *DEMeter*, Université de Lille 3. [hal-01257522](https://hal.archives-ouvertes.fr/hal-01257522)

COUPRIE, P. 2016. "Le développement d'un outil d'aide à l'analyse musicale : bilan et perspectives musicologiques". In *Musiques électroacoustiques. Analyses – Écoutes*, dir. Nicolas Marty. Sampzon, Delatour, p 157-169.

DAHAN, Kevin. 2005a. "Quelques réflexions sur la logique d'interface pour la création musicale assistée par ordinateur". In *Actes des Journées d'Informatique Musicales*, Saint-Denis.

DAHAN, K. 2005b. *Domaines formels et représentations dans la composition et l'analyse des musiques électroacoustiques : systèmes dynamiques dans la création musicale assistée par ordinateur*. PhD, Université Paris 8, Saint-Denis.

DAHAN, K. 2007a. "Reconstructing *Stria*". In *International Computer Music Conference*, Copenhagen.

DAHAN, K. 2007b. "Surface Tensions : Dynamics of *Stria*". In *Computer Music Journal*, vol. 31, no 3, p 65-74. DOI [10.1162/comj.2007.31.3.65](https://doi.org/10.1162/comj.2007.31.3.65)

DAHAN, K. 2012. "For music, mind, and machines for an integrative musicology". In *Music, Mind, and Invention*, Ewing Township.

DELALANDE, François. 2013. *Analyser la musique, pourquoi, comment ?*. Paris, INA-GRM.

DELEDALE, Gérard. 1978. *Charles S. Peirce. Ecrits sur le signe*. Paris, Le Seuil. (rassemble, traduit et commente des texte extrait du *Collected Papers* de C. S. Peirce, Harvard University Press).

DERRIEN, Jean-Pierre, ODIARD, Patrick. 1985. *Philippe Manoury*. Les Cahiers de l'Ircam, coll. Compositeurs d'aujourd'hui, no 8. Paris, Ircam-Centre Pompidou.

DESAIN, Peter, HONING, Hendjan. 1993. "Letter to the editor : the mins of max : The mins of Max". In *Computer Music Journal*, vol 17, no 2.

DE SOUSA DIAS, Antonio. 2003. "Transcription de fichiers MusicV vers Csound au travers de Open Music". In *Journées d'Informatique Musicale*, Rennes.

DI COSMO, Roberto. 2017. "Software Heritage : pourquoi et comment construire l'archive universelle du code source". In *Bulletin de la société informatique de France*, no 10 (avril), p 67-77.

DI SCIPIO, Agostino. 1995a. "Inseparable models of materials and of musical design in electroacoustic and computer music". In *Journal of new music research*, vol. 24, no 1, p 34-50.

DI SCIPIO, A. 1995b. "Centrality of techné for an aesthetic approach on electroacoustic music". In *Journal of New Music Research*, vol. 24, no 4, p 369-383.

DI SCIPIO, A. 2000. "An analysis of Jean-Claude Risset's Contours". In *Journal of New Music Research*, vol. 29, no 1, p 1-21.

DODGE, Charles, JERSE, Thomas A. 1997. *Computer music : synthesis, composition and performance*. New York, Schirmer Books.

DONIN, Nicolas. 2015. "Introduction". In *Genèses musicales*, dir. N. Donin, A. Grésillon, J.-L. Lebrave. Paris, Presses Universitaires de Paris-Sorbonne.

DUCHEZ, Marie-Elisabeth. 1979. "La représentation spatio visuel du caractère grave-aigu et l'élaboration de la notion de hauteur de son dans la conscience musicale occidentale". In *Acta Musicologica*, vol. 51, no 1, p 54-73.

DUCHEZ, M.-E. 1980. "La représentation de la musique : information d'action et expression structurelle dans la représentation de la musique occidentale traditionnelle". In *Actes du XVIIIe congrès des sociétés de Philosophie de langue française*, Strasourg.

DUCHEZ, M.-E. 1981. "Description grammaticale et description arithématique des phénomènes musicaux : le tournant du IX^e siècle". In *Sprache und Erkenntnis im Mittelalter*, vol 13, no 2.

DUCHEZ, M.-E. 1983. "Des neumes à la portée : élaboration et organisation rationnelle de la discontinuité musicale et de sa représentation graphique, de la formule mélodique à l'échelle monocordale". In *Canadian University Music*, no 4, p 22-65.

DUCHEZ, M.-E. 1991. "L'évolution scientifique de la notion de matériau musical". In *Le timbre, métaphore pour la composition*, dir. J.-B. Barrière, Paris, Christian Bourgeois, p. 47-81.

DUFEU, Frédéric. 2010. *Mémoires artificielles et outils de création musicale : Conséquences de l'emploi des lutheries électroniques et numériques sur la conception et l'appréhension des musiques électroacoustiques*. PhD, Université Rennes 2, Rennes.

DUFEU, F. 2013. "Comment développer des outils généraux pour l'étude des instruments de musique numériques ? Un prototype en JavaScript pour l'investigation de programmes Max". In *Revue Francophone d'Informatique Musicale*, no 3.

DUFOURT, Hugues. 1980. "Logique du matériau (1970/1980, un bilan)". In *Le compositeur et l'instrument*. Paris, Ircam - Centre Georges-Pompidou.

DUFOURT, H. 1981. "L'artifice d'écriture dans la musique occidentale". In *Critique*, vol. 408, p 465-478.

DUFOURT, H. 1996. "La musique sur ordinateur : une sémantique sans sujet ?". In *La musique depuis 1945, Matériau, esthétique et perception?*, dir. H. Dufourt et Joël-Marie Fauquet. Bruxelles, Madraga.

DUFOURT, H. 2004 . "De la notation à l'ordinateur". In *Cahiers de médiologie*, no 18.

EAGLESTONE, Barry M., FORD, Nigel. 2008. "Are cognitive styles an important factor in design of electroacoustic music software ?". In *Journal of New Music Research*, vol. 37, no 1.

EBBEKE, Klaus. 1990. "La vue et l'ouïe – Problématique des partitions dans la musique électroacoustiques (en particulier chez Karlheinz Stockhausen)". In *Contrechamps*, no 11, p 70-79.

EMMERSON, Simon. 1986. "The relation of language to materials". In *The language of Electroacoustic Music*, dir. S. Emmerson. London, Macmillan, p 17-39.

EMMERSON, S., LANDY, Leigh. 2016. *Expanding the Horizon of Electroacoustic Music Analysis*. Cambridge, Cambridge University Press.

EVERAERT-DESMEDT, Nicole. 1990. *Le processus interprétatif : introduction à la sémiotique de Ch. S. Peirce*. Bruxelles, Mardaga.

FARNELL, Andy. 2007. "An introduction to procedural audio and its application in computer games". In *Audio mostly conference*, Ilmenau, p 1-31.

FEIRREIRA, G. M. D. S. 2001. "When Art, Science and Technology meet : the Strength, the Dilemma or the Curse of Electroacoustic Music ? ". In *International Conference : Music Without Walls? Music Without Instruments*. De Montfort University.

FOBER, Dominique, LETZ, Stéphane. 1999. "Les logiciels d'aide à la composition musicale". In *Rencontres Musicales Pluridisciplinaires*, p 13-19.

GENEVOIS, Hugues. 1999. "Geste et pensée musicale : de l'outil à l'instrument". In *Les nouveaux gestes de la musique*. Marseille, Parenthèses.

GERZSO, Andrew. 1984. "Reflections on Répons". In *Contemporary Music Review*, vol. 1, no 1, p 23-34.

GIUGNO DI, Giuseppe, KOTT, J., GERZSO, Andrew. 1981. "Progress Report on the 4X Machine and Its Use". In *International Computer Music Conference*, USA.

GONCALVES, Tanya. 2014. "An audio programming language informed by the theory of cognitive dimensions". In *eContact!*, vol. 17, no 3.

GOODY, Jacques. 1979. *La raison graphique. La domestication de la pensée sauvage*. Paris, Editions de Minuit.

GRABÓCZ, Márta. 2013. *Entre naturalisme sonore et synthèse en temps réel : images et formes expressives dans la musique contemporaine*. Paris, Éditions des archives contemporaines.

HAWORTH, Christopher. 2015. "Sound Synthesis Procedures as Texts : An Ontological Politics in Electroacoustic and Computer Music". In *Computer Music Journal*. Vol. 39, no 1, p 41-58. DOI [10.1162/COMJ_a_00284](https://doi.org/10.1162/COMJ_a_00284)

HAYNES, Stanley. 1981. "'Mortuos Plango, Vivos Voco": A Realization at IRCAM". In *Computer Music Journal*, vol. 5, no 4, p 22-24. DOI [10.2307/3679502](https://doi.org/10.2307/3679502)

HUGLO, M. 1990. "Bilan de 50 années de recherches (1939-1989) sur les notations musicales de 850 à 1300". In *Acta musicologica*, vol. 62, no 2-3, p. 224-259. DOI [10.2307/932634](https://doi.org/10.2307/932634)

JAFFE, David A. 1995. "Ten criteria for evaluating synthesis techniques". In *Computer Music Journal*, vol. 19, no 1, p 76-87. DOI [10.2307/3681301](https://doi.org/10.2307/3681301)

KVIFTE, Tellef, JENSENIUS, Alexander Refsum. 2006. "Towards a coherent terminology and model of instrument description and design". In *International Conference on New Interfaces for Musical*, Paris.

LACHENMANN, Helmut. 1991. "Quatre aspects du materiau musical et de l'ecoute". In *Inharmoniques*, no 8-9, traduction de J.-L. Leleu d'un texte paru dans *Revue Musicale Suisse*, no 6, 1983.

LALIBERTÉ, Martin. 1994. *Un principe de la musique électroacoustique et informatique et son incidence sur la composition musicale*. PhD, EHESS-Ircam, Paris.

LALIBERTÉ, M. 2013. "Émergence et développement de l'informatique musicale". In *Théories de la composition musicale au xx^e siècle*, dir. N. Donin et L. Feneyrou. Lyon, Symétrie, p 639-670.

LASKE, Otto. 1990. "The computer as the artist's alter ego". In *Leonardo*, vol. 23, no 1, p 53-66.

LASKE, O. 1992. "For a lack of a better word". In *Computer Music Journal*, vol. 16, no 2, p 6-7. DOI [10.2307/3680708](https://doi.org/10.2307/3680708)

LE BOEUF, Patrick. 2003. "FRBR : Un p'tit coin de paradigme". In *First IFLA Meeting of Experts on an International Cataloguing Code*, Frankfurt.

LEIPP, Emile. 1971. *Acoustique et musique*. Paris, Masson.

LEMAN, Marc. 1993. "Symbolic and subsymbolic description of music". In *Music processing*, dir. G. Haus. Oxford University Press, p 119-164.

LEMOUTON, Serge. 2012. *Vingt ans de pratique de la Réalisation en Informatique Musicale*. Mémoire de Master 2, Université Paris-Est Marne-la-Vallée.

LEMOUTON, S., GOLDSZMIDT, Samuel. 2016. "La préservation des oeuvres musicales du répertoire de l'Ircam : Présentation du modèle de Sidney et analyse des dispositifs temps réel". In *Journées d'Informatique Musicale*, Albi.

LERESCHE Françoise, LE BOEUF, P. 2012. *Fonctionnalités requises des notices bibliographiques, rapport finale*, trad. de *Functional requirements for bibliographic records : final report*, rapport de l'International Federation of Library Associations (IFLA).

LEROI-GOURHAN, André. 1964. *Le Geste et la Parole : Technique et langage*. Paris, Albin-Michel.

LE VOT, Gérard. 1997. "Du nombre silencieux à la mise en son des nombres". In *Musique et mathématiques*, dir. H. Genevois et Y. Orlarey. Lyon, Aleas.

LE VOT, G. 1999. La notation de la musique en occident médiéval (signe, geste et invention). In *Musique et notations*, dir. Y. Orliac. Lyon, Aléas, p 11-40.

LEVY, Fabien. 2004. *Complexité grammatologique et complexité aperceptive en musique : étude esthétique et scientifique du décalage entre la pensée de l'écriture et la perception cognitive des processus musicaux sous l'angle des théories de l'information et de la complexité*. PhD, EHESS, Paris.

LEVY, Pierre. 1992. *De la programmation considéré comme des beaux arts*. Paris, La découverte.

LEVY, P. 1992. *Quest-ce que le virtuel ?* Paris, La découverte.

LICATA, Thomas. 2002. *Electroacoustic music : Analytical perspectives*. Westport, Greenwood Press.

LINDEMANN, Eric, DECHELLE, François, SMITH, Benett, STARKIER, Michel. 1991. "The architecture of the ircam music workstation3". In *Computer Music Journal*, vol.15, no 3, p 41-49. DOI [10.2307/3680764](https://doi.org/10.2307/3680764)

LORRAIN, Denis. 1980. *Analyse de la bande magnétique de l'œuvre de Jean-Claude Risset "Inharmonique"*, Rapport IRCAM, no 26/80. Paris, Ircam - Centre Georges Pompidou.

LORRAIN, D. 2009. "Interpolations". In *Le calcul de la musique*, dir. L. Pottier. Saint-Étienne, Publication de l'université de Saint-Étienne.

MAGNUSSON, Thor. 2009. *Epistemic Tools : The Phenomenology of Digital Musical Instruments*. PhD, University of Sussex, Angleterre.

MALLOCH, Joseph, BIRNBAUM, David, SINYOR, Elliot. 2006. "Towards a new conceptual framework for digital musical instruments". In *Proceedings of the 9th International Conference on Digital Audio Effects*, Montreal p 49-52.

MALT, Mikhail. 2009. "La composition assistée par ordinateur, modèles et calcul, quelques éléments de réflexion". In *Le calcul de la musique*, dir. L. Pottier. Saint-Étienne, Publication de l'université de Saint-Étienne.

MALT, M., JOURDAN, Emmanuel. 2015. "La "BSTD" – Une représentation graphique de la brillance et de l'écart type spectral, comme possible représentation de l'évolution du timbre sonore". In *L'analyse musicale aujourd'hui*, dir. Hascher Xavier et Ayari Monder. Sampzon, Delatour, p 111-128.

MANNING, Peter. 2004. *Electronic and Computer Music*. New York, Oxford University Press.

MANNING, P. 2006. "The significance of techné in understanding the art and practice of electroacoustic composition". In *Organised Sound*, vol 11, no 1, p 81-90. DOI [10.1017/S1355771806000112](https://doi.org/10.1017/S1355771806000112)

MANOURY, P. 1987. Note de programme du concert du 25 avril 1987.

MANOURY, P. 1996. Note de programme du concert du 10 février 1996.

MANOURY, P. 1997. Note de programme du concert en janvier 1997

MANOURY, Philippe, COHEN-LEVINAS, Danielle. 1998. *La note et le son : écrits et entretiens (1981-1998)*. Paris, L'Harmattan.

MANOURY, P., BATTIER, Marc, BONARDI, Alain, LEMOUTON, Serge. 2003. *Les musiques électroniques de Philippe Manoury*. Paris, Ircam - Centre Georges Pompidou.

MANOURY, P. 2007. "Considérations (toujours actuelles) sur l'état de la musique en temps réel". In *Étincelle*, no 3, Paris, Ircam - Centre Georges Pompidou.

MANOURY, P. 2016. *Entretien personnel*. Paris.

MANOURY, P. 2017. *Entretien personnel*. Strasbourg.

MARINO, G., SERRA, M. H., et RACZINSKI, J. M. 1993. "The UPIC system : Origins and innovations". In *Perspectives of New Music*, vol 31, no 1, p 258-269. DOI [10.2307/833053](https://doi.org/10.2307/833053)

MARTY, Nicolas. 2015. "Introduction – Écoutes et analyses". In *Musiques électroacoustiques - Analyses – Ecoutes*. Sampzon, Delatour.

MATHEWS, Max V. 1963. "The digital computer as a musical instrument". In *Science*, vol. 142, no 3592, p 553-557.

MATHEWS, Max V. (with the collaboration of Joan E. MILLER, F. R. Moore, J. R. Pierce et J. C. Risset). 1969. *The technology of computer music*. Cambridge, MIT press.

MAY, Andrew. 2006. "Philippe Manoury's *Jupiter*". In *Analytical methods of electroacoustic music*, dir. M. Symony. London, Routledge.

McCARTNEY, James. 2002. "Rethinking the computer music language : SuperCollider". In *Computer Music Journal*, vol. 26, no 4, p 61-68. DOI [10.1162/014892602320991383](https://doi.org/10.1162/014892602320991383)

MEEÛS, Nicolas, 1991. "Apologie de la partition". In *Analyse musicale*, vol. 24, no 3, p 19-22.

MEEÛS, N. 2001. "Le statut sémiologique de l'analyse musicale". In *Actes du Troisième Congrès Européen d'Analyse Musicale*, Montpellier.

MEEÛS, N. 2001. "La partition". In *Musicae Scientiae*, vol. 5, no 2, p 73-78. DOI [10.1177/10298649010050S211](https://doi.org/10.1177/10298649010050S211)

MEEÛS, N., 2016 . "Notation, transcription, analyse". In *Musiques orales, notations musicales et encodages numériques*, dir. Sylvaine Leblond Martin. Paris, Les Éditions de l'Immatériel.

MENEGHINI, Matteo. 2003. "Stria by John Chowning : analysis of the compositional process". In *XIV Colloquium on Musical Informatics*, Florence, p 8-10.

MENEGHINI, M. 2007. "An analysis of the compositional techniques in John Chowning's Stria". *Computer Music Journal*, vol. 31, no 3, p 26-37. DOI [10.1162/comj.2007.31.3.26](https://doi.org/10.1162/comj.2007.31.3.26)

MIRANDA, Eduardo Reck. 1995. "An artificial intelligence approach to sound design". In *Computer Music Journal*, vol. 19, no 2, p 59-75. DOI [10.2307/3680600](https://doi.org/10.2307/3680600)

MOLINO, Jean. 1975. "Fait musical et sémiologie de la musique". In *Musique en jeu*, vol. 17, p 37-62.

MOORE, Adrian. 2009. "Knowledge and Experience in Electroacoustic Music: where is the common ground ?". (www.adrianmoore.co.uk/research/bristol2.pdf, 2017-07-27).

MORIN, Edgar. 2005. *Introduction à la pensée complexe*. Paris, Éditions du Seuil.

MORRISON, Landon. 2014. "Graphical Music Representations : A Comparative Study Based on the Aural Analysis of Philippe Leroux's *M.É.*". In *Electroacoustic Music Studies*, Berlin.

MOUNIN, Georges. 1970. *Introduction à la sémiologie*. Paris, Editions de Minuit.

MOUNIN, G. 1997. *La sémantique*. Paris, Payot & Rivages.

NADRIGNY, Pauline. 2014. *Musique et philosophie au XXe siècle*. Paris, Classique Garnier.

NATTIEZ, Jean-Jacques. 1975. *Fondements d'une sémiologie de la musique*. Paris, Union Générale d'Édition.

NATTIEZ, J.-J. 1987. *Musicologie générale et sémiologie*. Paris, Christian Bourgeois.

NELSON, Goodman. 1990. *Langages de l'art. Une approche de la théorie des symboles*. Arles, Chambon.

NICOLAS, François. 2001. "Le problème de la double écriture". In *Analyse et création musicale*. Paris, L'Harmattan, p 223-253.

ORLAREY, Yann. 2009 . "Entre Calcul, programmation et création". In *Le calcul de la musique*, dir. Laurent Pottier. Presses Universitaires de Saint Etienne, p 331-365 .

PLESSAS, Peter, BOUTARD, Guillaume. 2015. "Transposition et interprétation de l'instrument de musique électronique composé". In *Journées d'Informatique Musicale*, Montréal.

POTTIER, Laurent. 2011. "'Turenas" de John Chowning, vers une version interactive". In *Musimediane*, no 6.

PUCKETTE, Miller. 1984. "The M Orchestra Language". In *International Computer Music Conference*, Paris.

PUCKETTE, M. 1988. "The patcher". In *International Computer Music Conference*, San Francisco, p 420-429.

PUCKETTE, M. 1991. "Combining event and signal processing in the MAX graphical programming environment". In *Computer music journal*, vol. 15, no 3, p 68-77. DOI [10.2307/3680767](https://doi.org/10.2307/3680767)

PUCKETTE, M. 1996. "Formant-based audio synthesis using nonlinear distortion". In *Journal of the Audio Engineering Society*, vol. 43, no 1/2, p 40-47.

PUCKETTE, M *et al.* 1998. "Real-time audio analysis tool for Pd and MSP". In *Proceedings of the International Computer Music*. San Francisco. p. 109-12.

PUCKETTE, M. 2001. "New Public-Domain Realizations of Standard Pieces for Instruments and Live Electronics". In *International Computer Music Conference*, Havana.

PUCKETTE, M. 2002a. "Max at seventeen". In *Computer Music Journal*, vol 26, no 4, p 31-43. DOI [10.1162/014892602320991356](https://doi.org/10.1162/014892602320991356)

PUCKETTE, M. 2002b. "Using Pd as a score language". In *Proceedings of the International Computer Music Conference*, Göteborg.

PUCKETTE, M. 2005. Pd Documentation, (msp.ucsd.edu/Pd_documentation/, 2017-07-25).

PUCKETTE, M. 2006. *The theory and technique of electronic music*. Singapore, World Scientific Publishing Co Inc.

RIOTTE, André. 1988. "Modèles et métaphores : les formalismes et la musique". In *La musique et les sciences cognitives*, dir. Steven McAdams et Irène Deliège. Bruxelles, Madraga.

RISSET, Jean-Claude. 1969. *An introductory catalogue of computer synthesized sounds*. Murray Hill, Bell Telephone Laboratories.

RISSET, J.-C. 1991. "Timbre et synthèse des sons". In *Le timbre métaphore pour la composition*, dir. Jean-Baptiste Barrière. Paris, Christian Bourgois.

RISSET, J.-C. 1992. "Le son numérique : une acoustique affranchie de la mécanique ?". In *Le Journal de Physique IV*, vol 2, no 1.

RISSET, J. C. 1996. "Calculer le son musical : un nouveau champ de contraintes ?". In *La musique depuis 1945 : matériau, esthétique et perception*, dir. H. Dufourt et J.M. Fauquet. Bruxelles, Madraga.

RISSET, J. C. 1998. "Un survol de l'informatique musicale". In *Rapport de mission Art-Science-Technologie*, pour le ministère de la Culture et de la Communication.

RISSET, J. C. 1999. "Evolution des outils de création sonore". In *Interfaces homme-machine et création musicale*, dir. H. Vinet et F. Delalande. Paris, Hermes, p 17-36.

RISSET, J. C. 2001. "Problèmes d'analyse : quelques clés pour mes premières pièces numériques, *Little Boy* et *Mutations*". In *Analyse en musique électroacoustique, Actes de l'Académie Internationale de Musique Electroacoustique de Bourges*. Bourges, Mnemosyme, p 169-177.

ROADS, Curtis. 1996. *The computer music tutorial*. Cambridge, MIT Press.

ROUSSEAU, Francis, BONARDI, A. 2002. "Will Semantic Web Mining in Music Revolutionize Musicology ?". In *Conferencia Iberoamericana de Inteligencia Artificial*, Sevilla.

ROY, Stephane. 2004. *L'analyse des musiques électroacoustiques : modèles et propositions*. Paris, L'Harmattan.

ROY, S. 2006. "L'analyse formelle en quête de significations profondes". In *Lien*, no IV (L'analyse perceptive des musiques électroacoustiques), p 16-19.

RUTZ, Hanns Holger. 2011. "Limits of Control". In *Sound and Music Conference*, Padova.

SAUSSURE, Ferdinand de. 1969. *Cours de linguistique générale*. Publié par Charles Bally et Albert Sechehaye avec la collaboration de Albert Riedlinger. Paris, Payot.

SCHAEFFER, Pierre. 1952. *A la recherche d'une musique concrète*. Paris, Éditions du Seuil.

SCHAEFFER, Pierre. 1966. *Traité des objets musicaux*. Paris, Éditions du Seuil.

SCHNELL, Norbert et Marc Battier. 2002. "Introducing composed instruments, technical and musicological implications". In *Proceedings of the 2002 conference on New interfaces for musical expression*, National University of Singapore, p 1-5.

- SEEGER, Charles. 1958. "Prescriptive and Descriptive Music-Writing". In *The Musical Quarterly*, vol. 44, no 2, p 184-195.
- SIMONI, Mary (Dir.). 2006. *Analytical Methods of electroacoustic music*. London, Routledge.
- SLOBODA, John A. 1988. *L'esprit musicien : la psychologie cognitive de la musique*. Bruxelles, Mardaga.
- SMITH, Julius O. 1991. "Viewpoints on the history of digital synthesis". In *Proceedings of the International Computer Music Conference*, McGill University.
- SOLOMOS, Makis. 1998. "Le devenir du matériau musical au XXème siècle". In *Cahiers de Philosophie du Langage*, no 3 (Musique, rationalité, langage. L'harmonie : du monde au matériau), dir. A. Soulez et H. Vaggione. Paris, L'Harmattan, p 137-151. [hal-01202919](#)
- SOLOMOS, M. 2003. "De l'opérateur". In *Formel, informel : musique-philosophie*, dir. M. Solomos et A. Soulez. Paris, Harmattan.
- SOLOMOS, M. 2008. "Les "opérations mentales de la composition" (Xenakis)". In *Intellectica - La revue de l'Association pour la Recherche sur les sciences de la Cognition (ARCo)*, vol. 2008/1-2, no 48-49, p 207-220. [<hal-00770202>](#)
- SOLOMOS, M., MERIC, Renaud. 2011. "Ecosystèmes audibles et structures sonores émergentes dans la musique d'Agostino Di Scipio. Une collaboration entre philosophie de la musique et analyse musicale". In *Musurgia : analyse et pratique musicales*, vol. 18, no 3, p 39-56. DOI [10.3917/musur.113.0039](#)
- SOLOMOS, M. 2013. *De la musique au son. L'émergence du son dans la musique des XXe et XXIème siècles*. Rennes, Presses universitaires de Rennes
- SOUSA DIAS, Antonio de. 2007. "Deux contributions à la pédagogie de la musique électroacoustique et de l'informatique musicale". In *Journée d'Informatique Musicale*, Lyon.
- STOCKHAUSEN, Karlheinz. 1973. "Musique et graphique". In *Musique en jeu*, no 13.
- STOCKHAUSEN, K. 1988. "Musique électronique et musique instrumentale". In *Contrechamps*, no 9.
- STRAYER, Hope. R. 2013. "From Neumes to Notes : The Evolution of Music Notation". In *Musical Offerings*, vol. 4, no 1. DOI [10.15385/jmo.2013.4.1.1](#)
- STROPPIA, Marco. 1984. "The analysis of electronic music". In *Contemporary Music Review*, vol. 1, no 1, p 175-180.

STROPPA, M. 1999. "La Composition Assisté par Ordinateur". In *Causeries sur la Musique, entretiens avec des compositeurs*, interview réalisée par Danielle Cohen-Levinas. Paris, L'Harmattan, p 63-88.

STROPPA, M., Lemouton, S., AGON, Carlos. 2002. "Omchroma ; vers une formalisation compositionnelle des processus de synthèse sonore". In *Journées d'Informatique Musicale*, Marseille.

SZENDY, Peter. 1998. "Musique, temps réel". In *Résonance*, no 14, Ircam - Centre Georges Pompidou.

THIEBAUT, Jean-Baptiste. 2010. *Sketching music : representation and composition*. PhD, University of London.

THOMAS, Jean-Christophe, MION, Philippe, NATTIEZ, J-J. 1982. *L'envers d'une oeuvre : "De natura sonorum" de Bernard Parmegiani*. Paris, Buchet-Chastel.

TIFFON, Vincent. 2006. "La représentation phonographique est-elle une aide pour l'analyse perceptive de la musique électroacoustique ?". In *Lien*, no IV (L'analyse perceptive des musiques électroacoustiques), p 3-13.

TSOUKALAS, Kyriakos. 2001. "A collaboration workflow from sound-based composition to performance of electroacoustic music using Pd as a framework". In *Pure Data Convention*, Weimar.

VAGGIONE, Horacio. 1995. "Objets, représentations, opérations". In *Ars Sonora*, no 2, p 33-51.

VAGGIONE, H. 1998. "Son, temps, objet, syntaxe. Vers une approche multi-échelle dans la composition assistée par ordinateur". In *Cahiers de Philosophie du Langage*, no 3 (Musique, rationalité, langage. L'harmonie : du monde au matériau), dir. A. Soulez et H. Vaggione. Paris, L'Harmattan, p 169-202.

VAGGIONE, H. 2001. "Some Ontological Remarks about Music Composition Processes". In *Computer Music Journal*, vol 25, no 1, p 54-61. DOI [10.1162/014892601300126115](https://doi.org/10.1162/014892601300126115)

VAGGIONE, H. 2003. "Composition musicale et moyens informatiques : questions d'approche". In *Formel, informel : musique-philosophie*, dir. M. Solomos et A. Soulez, Paris, Harmattan, p 91-116.

VAGGIONE, H. 2008. "Composition musicale : représentations, granularités, émergences". In *Intellectica*, vol. 1, no 2, p 155-174.

VAGGIONE, H. 2010. "Représentations musicales numériques : temporalités, objets, contextes". In *Manières de faire des sons*, dir. A. Soulez et H. Vaggione. Paris, L'Harmattan, p 45-82.

VARELA, Francisco J. 1988. *Invitation aux sciences cognitives*. Paris, Le Seuil.

VARENNE, Franck. 2009. *Qu'est ce que l'informatique*. Paris, Vrin.

VERFAILLE, Vincent, GUASTAVINO Catherine, TRAUBE, Caroline. 2006. "An interdisciplinary approach to audio effect classification". In *Conference on Digital Audio Effects (DAFx-06)*, Montreal, Canada.

VERMESCH, Pierre. 1998. "De la notation musicale à sa représentation mentale : la mémorisation des partitions chez les pianistes". In *Musique et notations*, dir. Y. Orlarey. Lyon, Aléas.

VINET, Hugues. 2003. "The representation levels of music information". In *International Symposium on Computer Music Modeling and Retrieval*, Berlin, p 193-209.

VON BERTALANFFY, Ludwig. 1968. *General systems theory*. New York, George Braziller.

WANG, Ge. 2008 . *The ChuckK audio programming language. "A strongly-timed and on-the-fly environ/mentality"*. PhD, Princeton University.

WIGGINS, Geraint, MIRANDA, Eduardo, SMAILL, Alan, HARRIS, Mitch. 1993. "A framework for the evaluation of music representation systems". *Computer Music Journal*, vol. 17, no 3, p 31-42. DOI [10.2307/3680941](https://doi.org/10.2307/3680941)

WILL, Udo. 1999. "La baguette magique de l'ethnomusicologue : Repenser la notation et l'analyse de la musique". In *Cahiers de musiques traditionnelles*, no 12, p 9-33.

WINOGRAD, Terry, FLORES, Fernando. 1994. *Understanding computers and cognition : A new foundation for design*. Bristol, Intellect Books.

XENAKIS, Iannis. 1963. *Musiques formelles*. Numéro spécial de *La revue musicale*, no 253-254. Paris, Richard-Masse.

ZATTRA, Laura. 2003. *Science et technologie comme sources d'inspiration au CSC de Padoue et à l'IRCAM de Paris*. PhD, Université Paris IV.

ZATTRA, L. 2005. "Analysis and analyses of electroacoustic music". In *Sound and Music Computing*, vol. 36, Salerno (Italy).

ZATTRA, L. 2015. "Génétiques de la computer music". In *Genèses musicales*, dir. N. Donin, A. Grésillon, J.-L. Lebrave. Paris, Presses Universitaires de Paris-Sorbonne.

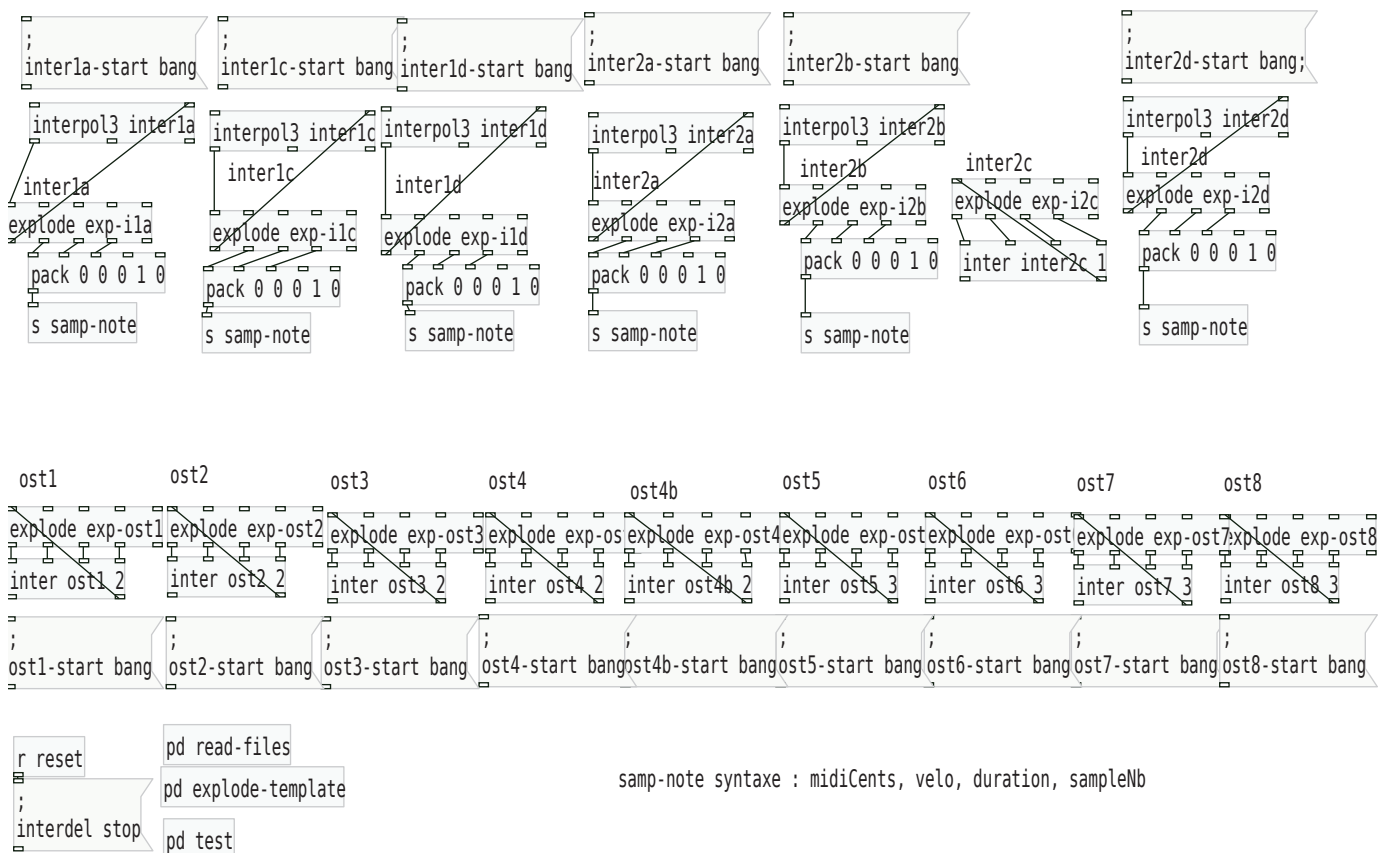
ZICARELLI, David. 1998. "An extensible real-time signal processing environment for Max". In *International Computer Music Conference*, Ann Arbor (États-Unis).

Annexes

Annexe A

L'algorithme d'interpolation, captures d'écran

Le sous-patch *interpol*



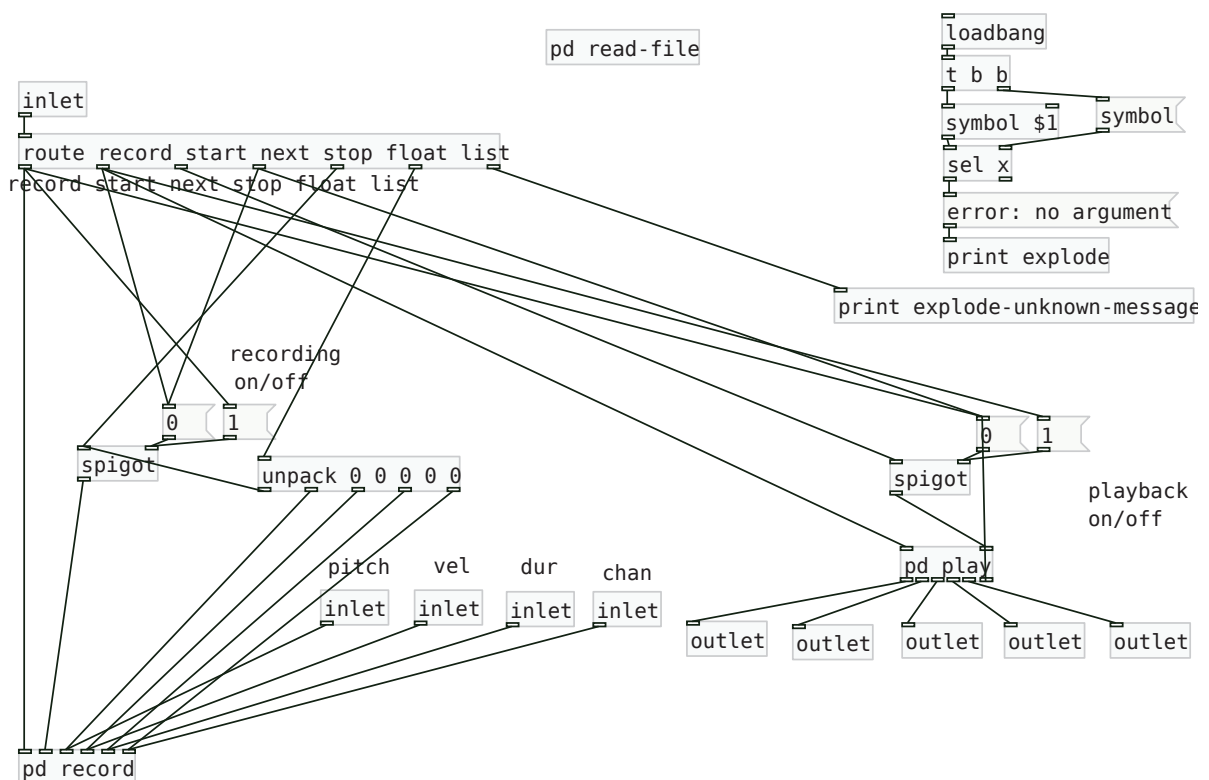
Le sous-patch *read-files*

```
pd exp-ila
pd exp-ilc
pd exp-ild
pd exp-i2a
pd exp-i2b
pd exp-i2c
pd exp-i2d
```

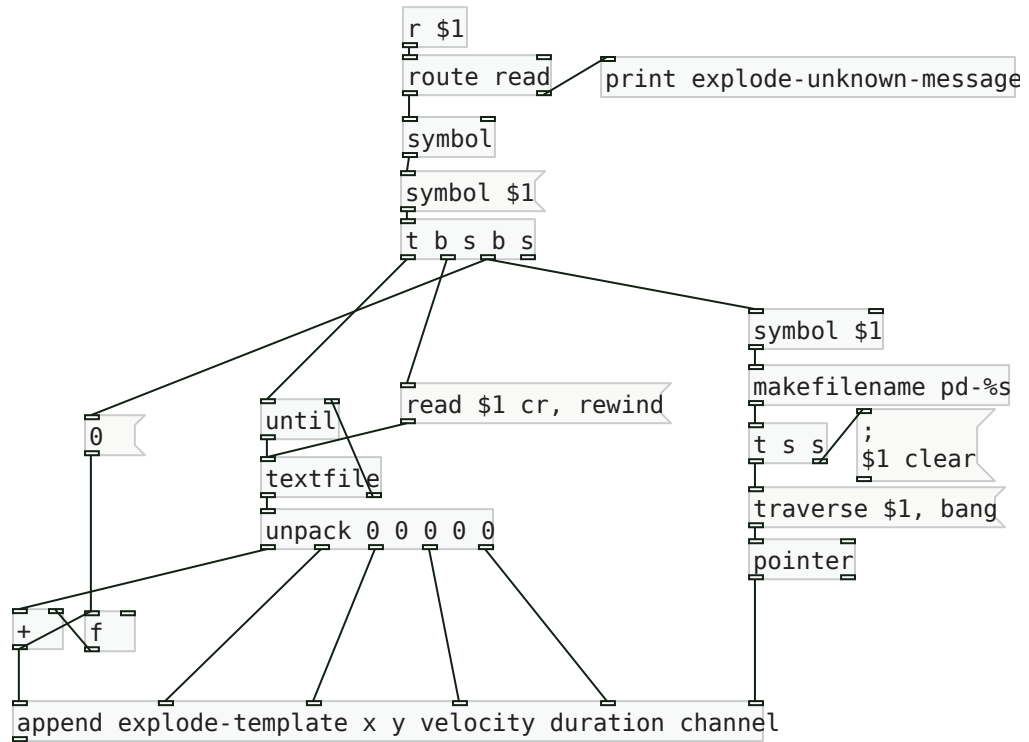
```
pd exp-ost1
pd exp-ost2
pd exp-ost3
pd exp-ost4
pd exp-ost4b
pd exp-ost5
pd exp-ost6
pd exp-ost7
pd exp-ost8
```

```
loadbang
;
exp-ila read interla.txt;
exp-ilc read interlc.txt;
exp-ild read interld.txt;
exp-i2a read inter2a.txt;
exp-i2b read inter2b.txt;
exp-i2c read inter2c.txt;
exp-i2d read inter2d.txt;
exp-ost1 read ost1.txt;
exp-ost2 read ost2.txt;
exp-ost3 read ost3.txt;
exp-ost4 read ost4.txt;
exp-ost4b read ost4b.txt;
exp-ost5 read ost5.txt;
exp-ost6 read ost6.txt;
exp-ost7 read ost7.txt;
exp-ost8 read ost8.txt;
;
exp-ila read interla.txt
;
exp-ost1 read ost1.txt
```

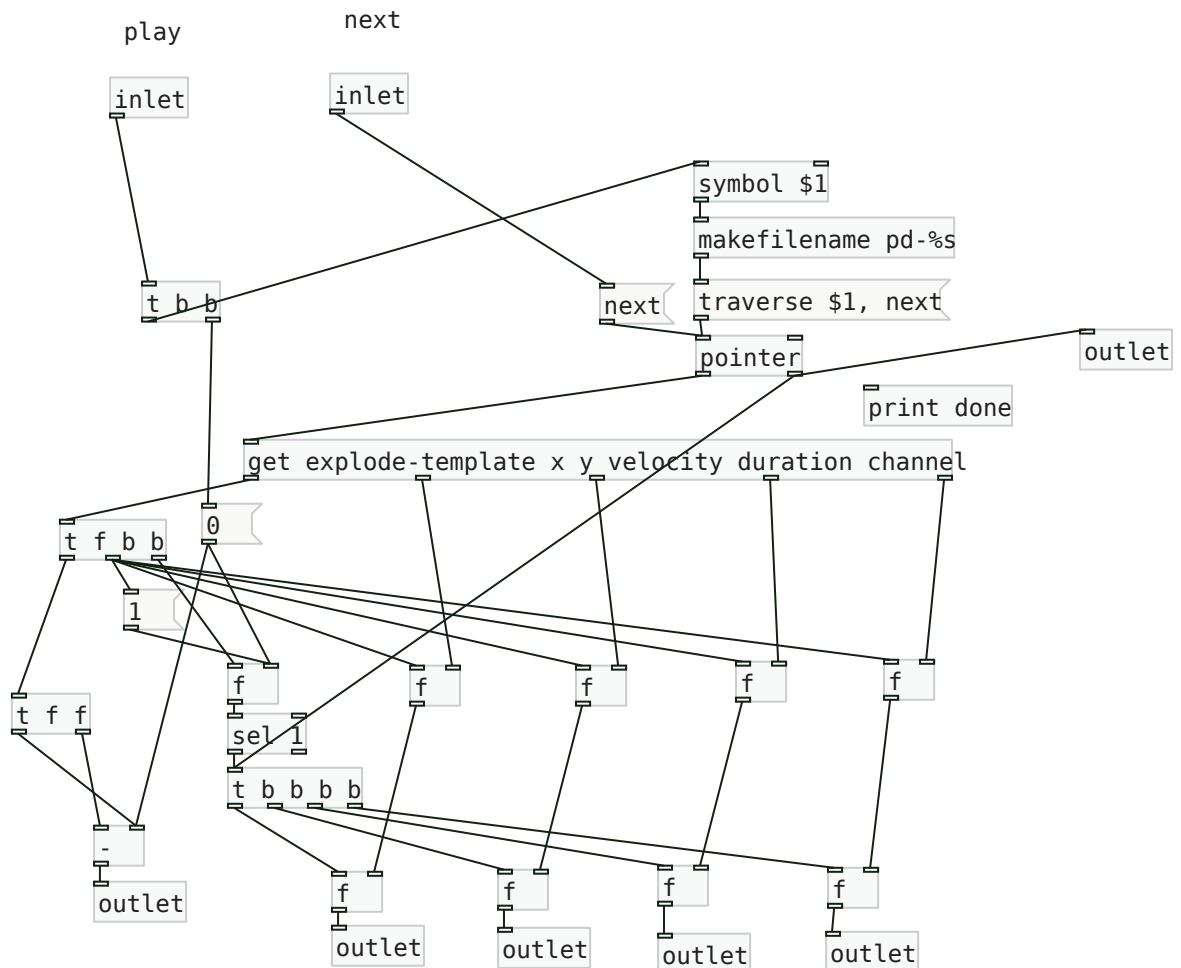
L'abstraction *explode*



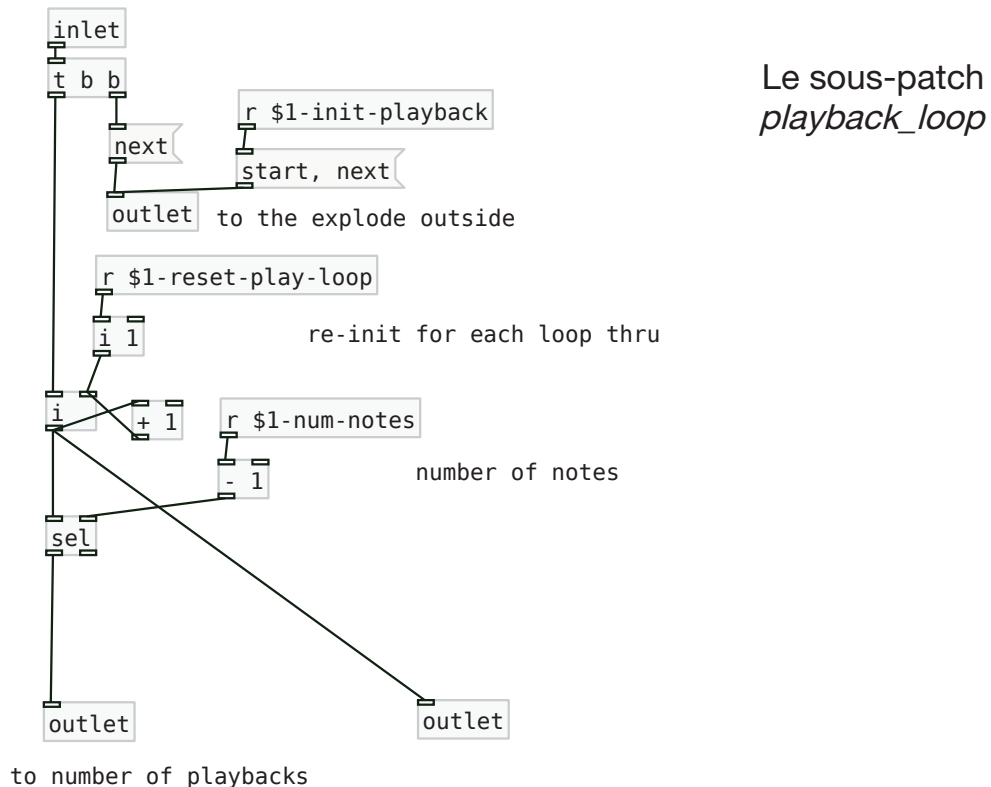
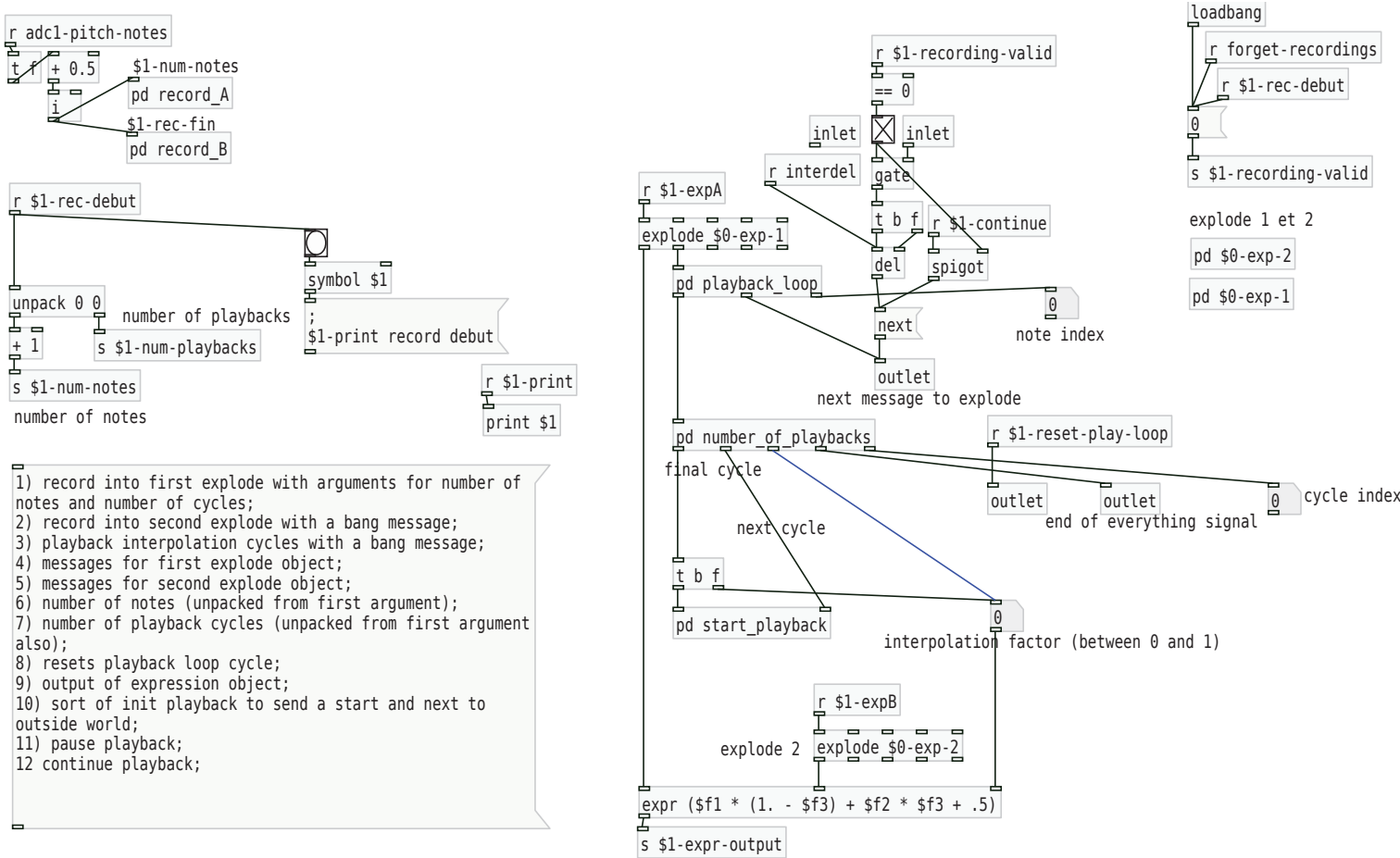
Le sous patch *read-file*



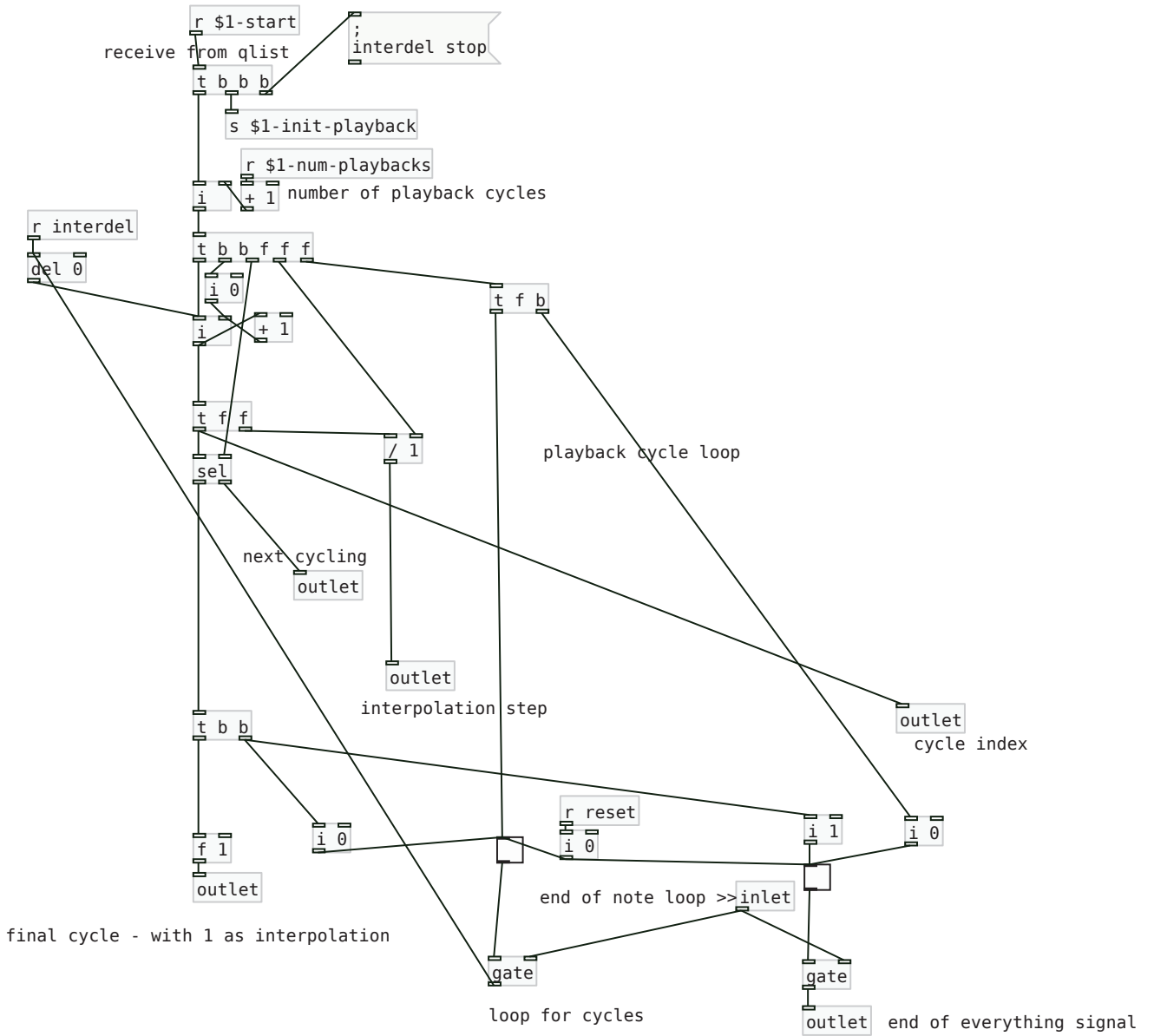
Le sous patch *play*



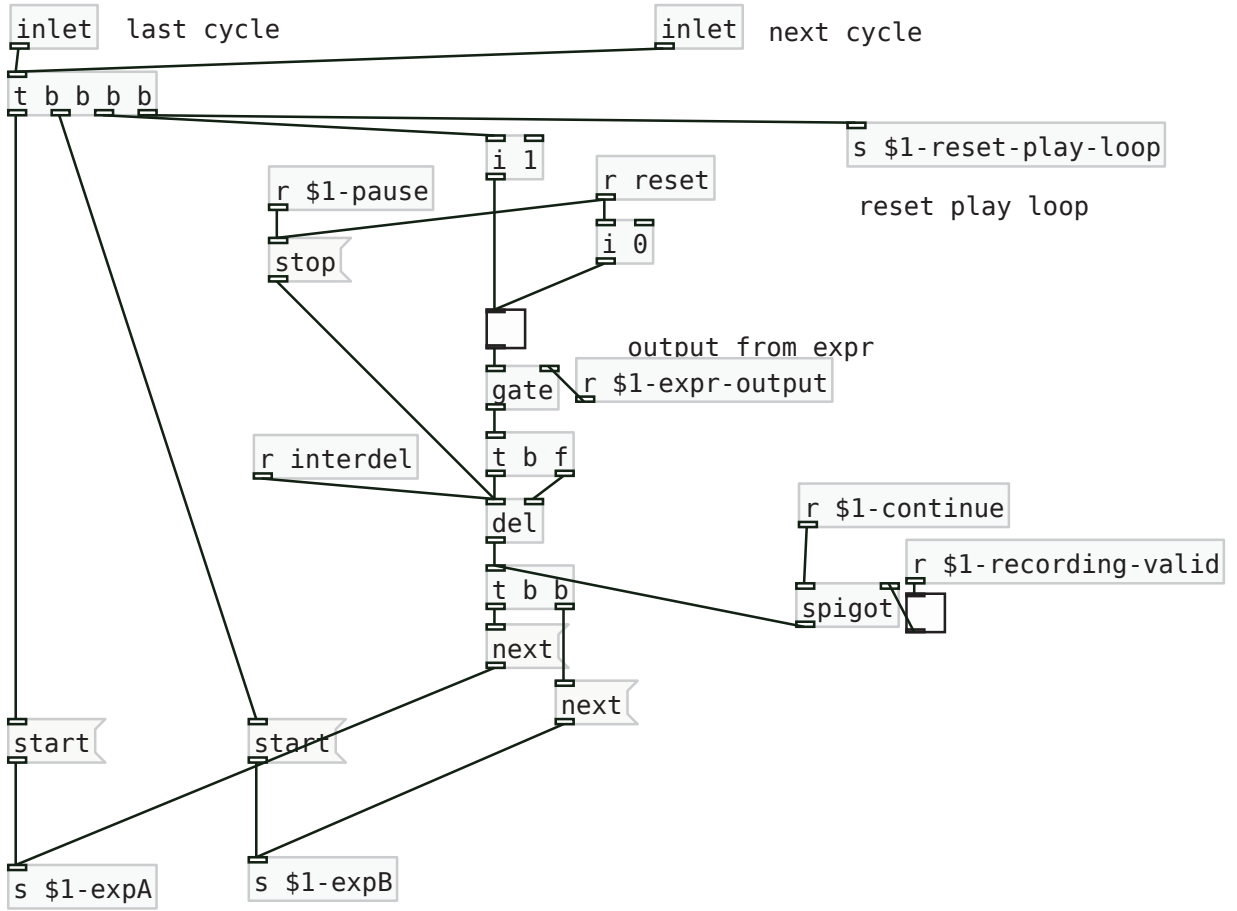
L'abstraction *interp3*



Le sous-patch *number_of_playbacks*



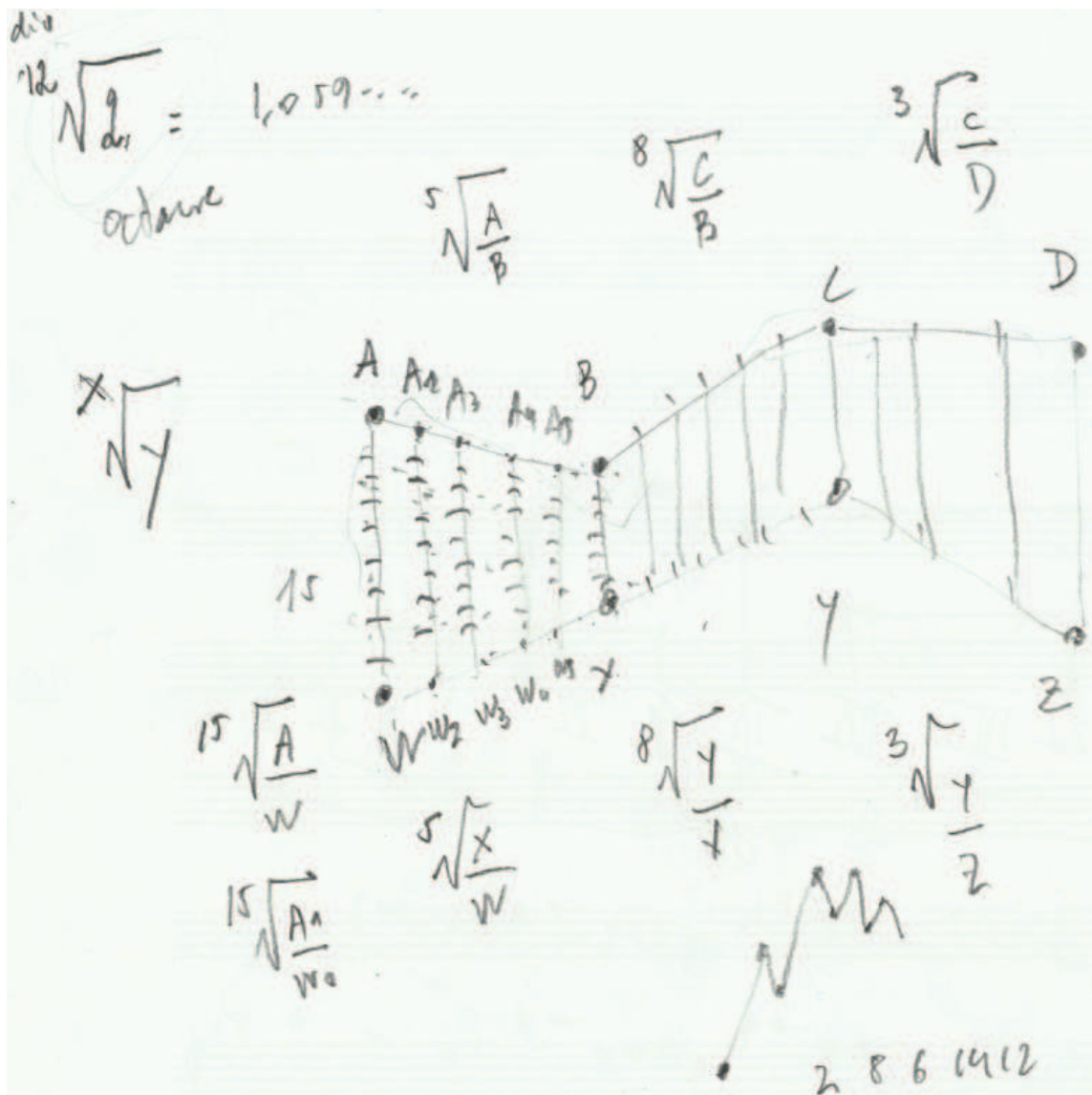
Le sous-patch *start_playback*



Annexe B

Notes pour l'algorithme d'interpolation du Sampler.

Extrait de [Manoury, 2017]



Annexe C

Description de l'algorithme qui extrait les chaînes de routing

1 Description textuelle

L'algorithme traverse les Modules du routing jusqu'à atteindre la fin d'une chaîne. Tous les modules traversés sont placés dans la variable *modsChains*. Une fois que l'algorithme se trouve dans une instance d'un module, la suite de ses opérations dépend des connexions de cette instance :

- si aucune connexion n'est présente – *i.e.* c'est une fin de chaîne –, alors la chaîne est enregistrée dans *routingChains*, puis l'algorithme se déplace au premier Module présent dans la variable *remain*.

- si des connexions sont présentes, alors deux données sont placées ensembles dans la variable *remain* : (i) le nom de l'instance dans laquelle l'algorithme se trouve et (ii) toutes les connexions sauf la première que l'instance possède. Ensuite l'algorithme se rend à cette première connexion.

remain est donc une variable composite, elle associe deux données de type String.

2 Description en pseudo-code

Voici une explication en pseudo-code des étapes réalisées par l'algorithme.

```
findRoutingChains(nomDuModule)
iterate = true
through = nomDuModule
tant que iterate est égal à true
    ajouter through a modsChain
    placer toutes les sorties du module dans un buffer
    si ce buffer est nul
        ajouter la chaîne analysée à routingChains
        si remain est nul, alors iterate = false
        si remain n'est pas nul
            effacer modsChain
            retrouver la chaîne où était présent le premier module de
remain (remain{0})
            charger tous les noms de modules présents dans cette
chaîne jusqu'à remain{0}
            trouver les modules connectés à remain{0}
            si ce nombre de modules est plus grand que 1, alors
attacher à remain{0} tous les autres modules sauf le premier.
            si ce nombre de modules est égal à 1, alors retirer
remain{0} de remain
            through = le premier module connecté à remain{0}

    si le buffer n'est pas nul
        pour toutes les autres connexions sauf que la première, les
placer dans remain en les attachant à through
        through = la première connexion
```


Annexe E

Indications textuelles dans la partition

Les indications textuelles de la partition sont données ci-dessous. Elles sont précédées de la sous-section et du numéro de la page.

- 1A p1 Attendre l'extinction de l'accord avant d'enchaîner.
- 1F p3 Les petites notes très rapides
- 2A p6 Record Début Inter. 1A.
- 2A p6 Record Fin Inter. 1A.
- 2A p6 Le rallentando n'affecte pas les petites notes qui doivent être jouées toujours rapidement.
- 2A p6 Record Début Inter. 1C.
- 2A p6 Record Fin Inter. 1C.
- 2A p6 Record Début Inter. 1D.
- 2A p7 Record Fin Inter. 1D.
- 3C p9 Les [poitns d'orgues] doivent avoir des durées inégales.
- 3F p10 Libre, détendu, quasi improvisando
- 3F p11 Attendre l'extinction de la partie synthétique avant d'enchaîner.

- 3G p14 Attendre environ 2" avant d'enchaîner
- 4B p16 Baisser l'amplification de la flûte
- 4B p16 Glissandi avec l'embouchure
- 4B p16 Remonter l'amplification de la flûte
- 5A p17 Jouer les événements 1 à 5 ad lib. pendant la durée de la séquence (approx. 40")
- 5A p17 Attendre la fin de la séquence pour jouer 6
- 5B p17 Jouer les événements 7 à 10 en suivant les variations d'ambitus de la séquence
- 5C p18 Les durées des [points d'orgues] doivent être variables
- 5C p18 Les actions GO provoquent la continuation de la séquence et figent le premier événement dans la réverbération. Les actions STOP arrêtent le déroulement de la séquence et figent le dernier événement dans la réverbération.
- 5C p18 Attendre la fin de séquence avant d'enchaîner.
- 6Q p21 Espacé
- 7A p22 Record Début Inter. 2A
- 7A p22 Record Fin Inter. 2A
- 7B p22 Record Début Inter. 2B
- 7B p22 Record Fin Inter. 2B
- 7C p22 Record Début Inter. 2C
- 7C p22 Record Fin Inter. 2C
- 8D p24 Toutes les [croches] pas trop brèves
- 8F p28 La séquence comprise entre 24 et 25 dure 29"
- 8F p28 Petit à petit irrégulier
- 8F p28 de plus en plus irrégulier
- 8F p28 Très irrégulier
- 8F p28 Attendre la fin de la séquence avant 28
- 8F p28 Enchaîner Xla

9A p29 Le trémolo doit commencer imperceptiblement et se prolonger après la séquence

9B p29 Les durées [point d'orgue] ne doivent être égales.

9C p29 enchaîner Xa

12H p33 Très lent et espacé

12H p33 Sans expression immobile

12H p33 Encore plus espacé

12H p33 Un peu plus serré....

12H p33 Très espacé et lent