

HABILITATION À DIRIGER DES RECHERCHES

présentée à

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par

GÉRARD ASSAYAG

Spécialité : INFORMATIQUE

**Algorithmes, langages et modèles pour la recherche
musicale : de la composition à l'interaction**

Soutenue le 5 juin 2009 à l'Ircam, Paris

Après avis de :

Jean-Pierre Briot, DR CNRS, UPMC, rapporteur

Philippe Codognet, professeur, UPMC / CNRS, rapporteur

Stephen McAdams, DR CNRS, professeur, McGill University, rapporteur

Devant la commission d'examen formée de :

Michèle Castellengo, DR émérite CNRS, UPMC

Philippe Codognet, professeur, UPMC / CNRS

Myriam Desainte-Catherine, professeur, Université de Bordeaux 1

Stephen McAdams, DR CNRS, professeur, McGill University

Camilo Rueda, professeur, Pontificia Universidad Javeriana

*A peine franchie, sous les nuées, cette sombre ligne de
faîte, tout le pays, en contrebas, dispense des reflets.*

Jean Ricardou, *Les lieux-dits*.

Toute ma gratitude va à Myriam Desainte-Catherine dont l'énergie amicale a été déterminante pour ce travail d'habilitation.

A mes parents Jacqueline et Victor, qui m'ont guidé, à mes frère et sœur Patrick et Nathalie, à Nicole qui m'a toujours soutenu, à nos enfants Noam et Mona,

A Claudy l'initiateur, à Carlos, el compañero d'aventure, à Dan qui aurait dû voir cela, à toute l'équipe Représentations Musicales passée et future (avec Karim) qui m'a tant appris, et, bien sûr, aux inestimables OMax Brothers, Marc, Georges et Shlomo.

TABLE DES MATIERES

Introduction	7
Prologue : qu'est-ce qu'un ordinateur ?	11
Paradigmes et représentations	12
Logique, non ?	13
Computer / Composer	15
Premiers pas à l'Ircam	16
Nouvelles techniques instrumentales	16
L'attente d'un environnement pour la composition	17
MuScript, un langage pour la notation musicale	18
Crime, un environnement pour la composition	21
Développements	23
Boards	24
Retour à l'Ircam : de PatchWork à OpenMusic	28
Fondation de l'équipe Représentations Musicales : stratégies de recherche musicale	29
OpenMusic	30
Protocole dynamique	33
Programmation fonctionnelle	33
Programmation par objets	34
Programmation par contraintes	34
MOP Graphique	35
Maquettes	35
Temps et calcul dans OpenMusic	38
OpenMusic et Java : jOpenMusic	39
Applications pédagogiques d'OpenMusic	40
Un support expérimental pour la recherche musicale	41
Modélisation d'œuvres et de systèmes	41
Modèle informatique d'une pièce pour clarinette d'Igor Stravinsky.	41
Programmation par contraintes et harmonie tonale	42
Imagerie musicale pour l'analyse harmonique : OMiel	43
Extraction de la pulsation, de la métrique et des motifs polyphoniques	45
Analyse computationnelle de partitions selon un modèle d'écoute	47
Segmentation et reconnaissance de patterns : perception, analyse, modélisation.	49
Mathématique et Musique : approche algébrique des théories musicales	51

Du rythme	55
KANT : Une critique de la quantification pure.	55
Opérateurs sur les structures rythmiques dans OpenMusic	56
Extraction de la pulsation et de la métrique en temps-réel	57
Le geste dans OpenMusic	59
Intégration signal-symbolique	59
Composer la synthèse	59
Agencer la synthèse comme un espace	61
Représenter les données d'analyse	63
Ecrire le son et la synthèse	65
Orchestration computationnelle	66
Contraintes et processus	69
Objets , contraintes et processus	69
Contraintes et recherche locale	71
Programmation concurrente avec contraintes	72
Apprentissage et interaction	74
Interaction texte et musique	74
Apprentissage du style musical pour l'interaction	75
Partitions interactives	81
OMax : modélisation du style, de l'interaction, de l'improvisation	82
Conclusion et perspectives	95
Sélection de pièces utilisant OpenMusic	98
Publications personnelles / sous ma direction	100
Journaux, chapitres de livres et livres	100
Conférences avec comité de programme	103
Rapports	107
Travaux d'étudiants encadrés	107
Logiciels	110
Bibliographie du domaine	112

Dans le texte qui suit, les références bibliographiques concernant mes publications personnelles et celles des étudiants sous ma direction directe sont entre crochets, celles qui concernent des travaux issus de mon équipe sont entre crochets et en grisé et les références générales sont entre parenthèses.

1. Introduction

Le présent mémoire décrit mes travaux dans la période qui va de 1985 à aujourd'hui. C'est une période inhabituellement longue pour une habilitation, mais cela est lié à mon histoire personnelle. Ma carrière de chercheur s'est entièrement déroulée à l'Ircam (Institut de Recherche et de Coordination Acoustique Musique) où elle s'est déclinée en quatre phases :

1. Introduction des premiers systèmes de composition assistée par ordinateur à l'Ircam dans les années 80.
2. Travaux sur les environnements de programmation multi-paradigmes : fonctionnel, objet, contraintes, visuel ayant abouti à OpenMusic dans les années 90.
3. Travaux sur la modélisation du style par des méthodes issues de la théorie de l'information et de l'apprentissage automatique dans les années 2000.
4. Travaux récents sur l'interaction improvisée musicien-machine, au-delà des effets sonores, par intégration d'un modèle stylistique du musicien
5. Vers des environnements à échelles temporelles multiples, intégrant la notion de composition et d'interaction.

Je commence par le dernier point, le plus récent, car il m'apparaît, au moment où je m'efforce de rassembler et de justifier les travaux accumulés pendant plusieurs années, qu'il constitue une tentative de synthèse de ces travaux — une synthèse devenant possible après avoir mené ces derniers à terme.

Composer et *interagir* constituent probablement l'alpha et l'oméga de l'informatique musicale depuis ses débuts dans les années cinquante. Une grande partie de mon travail est fondée sur une critique des approches classiques de ces deux activités, la première étant souvent conçue comme la substitution d'un algorithme à la tâche artisanale du compositeur, et la seconde étant souvent limitée à une activité réflexe (et non pas réflexive) de la part de l'ordinateur, ce qui en musique revient à se limiter à la fonction d'ornementation. J'ai donc approché ces deux catégories à ma manière.

Dans le premier cas (composer), j'ai privilégié une approche architecturale, essentiellement hors-ligne, dans laquelle trois catégories du temps sont à l'œuvre : le temps, abstrait et représenté plutôt comme une dimension spatiale, de la partition (quel que soit le sens que l'on donne à ce codage extensif de l'œuvre musicale) ; le temps du compositeur, qui travaille par étapes et touches rétrospectives ou prospectives, visitant ce temps de la partition selon un parcours complexe et en tout état de cause ra-

rement causal ou chronologique, et interagissant avec des algorithmes informatiques ; le temps informatique, celui du déroulement des algorithmes de génération et de transformations de structures musicales

Dans le second cas (interagir), il s'agit d'une approche essentiellement discursive, en ligne, avec une forte (mais non exclusive) composante temps-réel. De nouveau, trois catégories de temps sont à l'œuvre : le temps du musicien, en particulier du musicien improvisateur, se déroule selon un schéma causal, du moins pour ce qui concerne la manifestation phénoménologique du jeu musical, disons, vu de l'ordinateur, un signal sonore ; le temps de ce que j'appelle l'avatar du musicien, un modèle informatique, élaboré incrémentalement en fonction de la captation du musicien, qui se scinde en deux régimes temporels complexes : celui de la décision (quoi jouer, en fonction de tout le passé analysé et stocké dans le modèle) et celui l'anticipation (ce qui est effectivement planifié comme devant être joué, avec une certaine portée dans le futur, mais qui est susceptible d'être déjoué, lorsque c'est encore possible).

Composer et interagir, quel que soit le sens qu'on accorde à ces termes, relèvent de deux modes cognitifs bien distincts qui ont engendré des travaux spécifiques en sciences et technologies de la musique et provoqué l'apparition de deux familles d'environnements informatiques qui ne se rejoignent pas : d'un côté, les systèmes de composition algorithmique ou d'assistance à la composition, dont OpenMusic est un exemple, de l'autre, les systèmes de traitement du signal et d'interaction temps réel, dont Max est un exemple. Vouloir intégrer composition et interaction, comme je le tente depuis quelques années, oblige à se poser la question de ces régimes temporels qu'il est difficile de faire marcher ensemble. Un certain nombre d'éléments issus de mes travaux récents ou moins récents permettent de poser la question à nouveaux frais.

Les travaux sur la modélisation stylistique, menés en collaboration avec Shlomo Dubnov de l'Université de San Diego, nous ont permis d'appliquer des techniques issues de la théorie de l'information pour capturer par un apprentissage simple les caractéristiques de surface d'un style musical, à partir d'une partition ou d'un enregistrement. Initialement, nous avons eu l'idée originale de considérer certaines représentations issues de la compression de données comme susceptibles de capturer de manière efficace et de structurer de manière ordonnée ces caractéristiques. Selon cette approche un modèle d'une séquence musicale est une représentation de cette dernière après réduction par un compresseur universel de type Lempel-ziv. Il a été montré que cette famille d'algorithmes modélise asymptotiquement (sur de longues séquences) des sources markoviennes dont l'ordre n'est pas a priori borné. Nous avons testé les variantes stylistiques créées par engendrement à partir de ces modèles dans le cadre d'un protocole de psychologie expérimentale dirigé par E. Bigand au Laboratoire d'Etudes de l'Apprentissage et du Développement (UMR 522 du CNRS). Les résultats ont montré que les sujets, à qui étaient présentées des musiques

originales et des artefacts, et dont on pouvait mesurer le temps qu'ils mettaient à les discriminer, pouvaient « se tromper » pendant plusieurs dizaines de secondes d'affilée. J'ai alors proposé avec Marc Chemillier d'appliquer ces résultats à des situations d'improvisation dans lesquelles un musicien joue avec un double de lui-même qui se constitue et se développe en temps réel. Les modèles de compression étant mal adaptés à une telle situation j'ai proposé d'utiliser une représentation dérivée de l'Oracle des Facteurs, une structure inventée par Maxime Crochemore et ses collègues au laboratoire d'informatique de l'Université de Marne-la-vallée, en montrant que cette dernière présentait les caractéristiques de source Markovienne d'ordre variable tout en permettant une représentation (quasi) complète des répétitions de motifs à chaque pas incrémental d'apprentissage. Ce résultat original découle du fait que l'Oracle est en effet équivalent à un arbre de suffixes complet qui a subi un « écrasement » de toutes ses branches le long de la branche la plus longue, le transformant en un graphe linéaire dont les flèches permettent, à partir de n'importe quel état, de sauter à la « continuation » en fonction du contexte commun le plus long (cette longueur correspondant en effet à un ordre markovien variable selon l'état considéré). Ce système, par nature adaptatif, a permis de mettre en œuvre les techniques de simulation stylistique en temps réel et de créer l'environnement d'improvisation avec ordinateur OMax, avec Georges Bloch et Marc Chemillier, un système maintenant opérationnel et utilisé dans des situations réelles de performance artistique (par exemple au Centre Pompidou le 8 juin 2009 avec le groupe Aka Moon, ou avec l'Ensemble Intercontemporain le 7 juin 2008 dans la création de Yann Robin *Art of Metal III*).

Ces recherches nous amènent à reconsidérer l'interaction : cette dernière, au lieu de se cantonner à une série d'actions réflexes de l'ordinateur en réponse au signal envoyé par le musicien, se développe maintenant selon sa logique propre, une logique liée non seulement aux signaux d'entrée, mais aussi au modèle du musicien — ou *avatar*, modèle de plus en plus riche au fur et à mesure que ce dernier apprend. Ceci conduit à des situations d'interaction tout à fait inédites. Bien entendu, les avatars peuvent être archivés, et le mécanisme d'apprentissage conduit à une hybridation très naturelle, lorsqu'il est conduit à partir d'un état archivé plutôt qu'à partir de zéro. Ainsi, si l'interaction d'un musicien avec son clone constitue l'expérience de référence, il est évidemment loisible de placer cette interaction dans un contexte où le modèle est issu d'un autre musicien, ou encore dans celui d'une hybridation de styles.

A partir de ces deux grands axes de recherche, la composition assistée par ordinateur, qui a conduit à l'environnement OpenMusic et l'improvisation avec un modèle du musicien, qui a conduit à l'environnement OMax, deux environnements opérationnels validés par des utilisateurs musiciens professionnels, nous arrivons au temps de la synthèse. Est-il possible d'imaginer un environnement qui gère à la fois

un texte composé, écrit — éventuellement à l'aide de procédures algorithmiques — et une interaction au sens évolué où nous l'entendons ?

Selon un modèle proposé par le compositeur Georges Bloch, dans le cas le plus général nous avons maintenant une entrée, le(s) musicien(s), un noyau de modélisation, qui construit l'avatar, une partie composée, le texte, qui constitue une donnée, et un noyau de génération, qui tient compte de l'avatar et du texte. La sortie du système est la séquence d'événements musicaux engendrés, à laquelle s'ajoute un flux de prescriptions symboliques (par exemple sous forme de partitions génératives) en direction des musiciens. Ces prescriptions constituent un nouveau type de sorties : elles infléchissent l'interprétation du texte par les musiciens, voire changent ce dernier. Les musiciens interprètent le texte, improvisent lorsque le texte — éventuellement modifié ou annoté pendant la session — le permet, l'ordinateur interprète des parties imposées s'il y en a, et improvise en activant l'avatar, mais cette fois-ci en respectant les contraintes issues de la présence d'un texte. Je propose de représenter le texte par des constructions symboliques déclaratives avec contraintes, exprimant à la fois ce qu'il contient de littéral et ce qu'il contient d'ouvert et modulable en fonction de l'évolution de la session. De même, le noyau de génération doit contraindre les élaborations de l'avatar en fonction des propriétés logico-temporelles imposées par le texte. Du point de vue des régimes temporels cela entraîne une série de problèmes intéressants. Notamment, une séquence générative peut être planifiée en cohérence avec l'état courant et passé du système et les contraintes musicales agissantes. Cette séquence peut être remise en question (c'est d'ailleurs un phénomène banal dans l'improvisation) un peu plus tard alors qu'elle a déjà été jouée en partie, en fonction d'entrées sonores imprévues. La réponse normale du système est alors de modifier le futur de cette séquence pour maintenir la cohérence. Cependant, on peut facilement arriver à un état surcontraint, et une des échappatoires est alors de modifier le texte, et de communiquer cette information aux musiciens (par exemple de manière visuelle, par un changement dans les directives ou dans la partition). De nouveau nous arrivons à une expérience tout à fait inédite dans l'interaction, mais dans un cadre élargi où la notion de composition est bien prise en compte. Cette recherche actuelle pose un certain nombre de difficultés, notamment dans le domaine des contraintes concurrentes, et nous avons dans cette perspective monté deux groupes de travail : le groupe de travail Contraintes, Musique et Interaction, de l'Association Française d'Informatique Musicale, 2008-2009, avec l'Ircam, le LINA, le LABRI, le LIFO, l'Universidad Javeriana de Cali ; le projet REACT (Robust theories for Emerging Applications in Concurrency Theory), une collaboration entre l'Ircam, l'AVISPA Research Group de l'Universidad Javeriana et l'équipe Comète du LIX, Inria/École Polytechnique.

Je termine cette introduction par quelques mots sur mes travaux en Composition Assistée par Ordinateur : ces derniers ont occupé une grande partie de mon temps et

abouti au logiciel OpenMusic dont une des caractéristiques importantes est qu'il est à notre connaissance le premier environnement de programmation visuel multi-paradigme (fonctionnel, objet, par contraintes) entièrement construit par des méthodes de méta-programmation. Cet environnement continue d'évoluer et constitue la base expérimentale pour toutes les recherches que nous avons évoquées. On détaillera un certain nombre de projets de recherche musicale adossés à OpenMusic. Mais avant de commencer, il n'est pas inutile, puisque tout notre passé de recherche est basé sur l'assomption que l'on peut utiliser l'ordinateur pour accompagner le processus créatif (suivant en cela une lecture de jeunesse d'Abraham Moles, *Art et ordinateur*, Casterman, Paris, 1971), de se demander ... qu'est-ce qu'un ordinateur.

2. Prologue : qu'est-ce qu'un ordinateur ?

Computeur, et non ordinateur. A la limite, calculeur. Ou alors, parler plutôt du champ : l'informatique (musicale). Mais qu'est-ce qu'une machinerie informatique qui se mêlerait de musique ? Un phonographe amélioré ? Un instrument relevant de l'organologie numérique ? Tout cela sans doute, mais pas seulement. Plutôt un être de langage, c'est-à-dire un être tout de langage, pas comme nous autres clivé d'affects. En tous cas pas un objet, une machine ou un outil. D'autant moins que le public est, lui, assailli d'objets baladeurs qui forment le système technique de la musique aujourd'hui, avec en toile de fonds le codage numérique, le logiciel, la mise en réseau. C'est la prolifération de ces prothèses numériques qui nous fait croire au ordinateur-machine-objet, comme si le langage, à travers les supports successifs de l'écriture, se réduisait à une matérialité d'argile, d'encre ou de cristaux liquides.

En d'autres termes, l'informatique possède une place singulière dans la sphère des techno-sciences : elle ne se tient pas à la construction d'objets imitant le corps (pelle / rétine) ou l'extrapolant (moteur), mais plutôt à la simulation des fonctions de l'esprit. Comme l'a bien formulé le philosophe Gérard Chazal (Chazal 1995), elle s'oppose à toute conception qui vise à séparer de manière radicale les formes de leurs contenus puisque ses constructions visent à une représentation des connaissances dans un cadre à la fois opérationnel et formel. De ce fait, l'informatique est un outil de la connaissance en même temps qu'un objet de connaissance, un réseau de significations où le sens percole, traversant des niveaux successifs de codage, où s'interconnectent le réseau interne du logiciel et le réseau global des machines.

Autant dire, un milieu rêvé pour la musique, bien au-delà de l'écume numérique marchandisée, bien plutôt au niveau de ce calcul secret leibnizien réactualisé par Jean-Claude Risset dans un article célèbre (Risset 1977).

2.1. Paradigmes et représentations

Il y a donc du calcul dans la musique, du code, du langage, de l'information, des structures, des régimes temporels, qui peuvent être décrits par des formalismes informatiques, dans un double jeu de l'analyse et de la synthèse. Le terme de modélisation peut alors s'appliquer, et se décliner selon divers paradigmes scientifiques.

Soit, si on s'intéresse à l'analyse :

- L'élaboration et l'implémentation informatique de théories musicales. Il s'agit d'une conceptualisation prospective, plus ou moins mathématisée, et qui, agissant comme outil de classification, peut permettre d'introduire des repères dans une matière musicale brute.
- La modélisation informatique d'œuvres musicales, pouvant procéder d'une théorie ad hoc, qui agit à la fois comme explicitation systémique de l'œuvre, et comme moteur de simulation permettant d'explorer paramétriquement son voisinage : l'intérêt pédagogique naît du fait que l'œuvre se définit aussi de qu'elle aurait pu être. La modélisation ne permet pas d'expliciter les choix subjectifs de l'auteur, mais elle a le mérite de les identifier (Malt 2001).
- L'analyse génétique informatique, qui cherche à tirer parti des traces numériques laissées par le compositeur utilisant l'informatique dans la phase de conception. Le résultat de l'analyse génétique informatique peut constituer un modèle « autorisé », issu du compositeur lui-même (Riotte 1999).
- L'utilisation d'outils de découverte (knowledge discovery). La musique considérée comme un médium hautement organisé, mais pour lequel on ne dispose pas de théorie a priori. Cette approche permettrait de faire émerger des représentations non-standard (visuelles, auditives, haptiques...) porteuses d'intelligibilité [Lartillot 2002b].
- Les outils basés sur des modèles cognitifs, prenant en compte la temporalité dans l'analyse, (et l'oubli...) comme le fait un sujet humain. Plutôt qu'une limitation, il faut voir la contrainte cognitive comme un filtre limitant l'explosion combinatoire inhérente aux approches purement formelles et mathématisantes (Leman 1997).

Si on s'intéresse à la synthèse :

- L'apprentissage automatique, par des moyens statistiques, issus de la théorie de l'information, par des systèmes dynamiques tels que les réseaux neuronaux ou les algorithmes génétiques. L'œuvre est prise comme la surface mouvante de processus internes qui se déploient de manière organique (Miranda 2007).

- L'approche syntaxique par les grammaires et les logiques formelles, approche encore fortement teintée de structuralisme. L'œuvre est considérée comme une instance déterminée d'un modèle formel plus ou moins rigide (Chemillier 1989).
- Les techniques issues de l'intelligence artificielle, aujourd'hui élargies au champ des sciences cognitives, par représentations des connaissances et inférences logiques. La modélisation de certains fonctionnements perceptifs et cognitifs chez le sujet musical (l'auditeur, le compositeur) est exploitée plutôt qu'une structure formelle de l'œuvre (Balaban & al 1992).

Bien sûr analyser et synthétiser/créer dans un contexte de contrôle formel informatique sont fortement réversibles. L'acte d'analyser est par nature un acte de transfert, entre objets, entre codes, entre systèmes de références. L'informatique se plaît à ce jeu, parce qu'elle est elle-même fondée sur la diffusion et la transduction de significations à travers des couches de codage qui dénotent des niveaux d'abstraction différenciés, parmi lesquels le codage graphique occupe une place privilégiée. D'où l'idée d'une analyse orientée vers le transfert visuel. Un recensement de ces espaces de représentation pour la musique reste à faire, afin de voir clair dans un champ de recherche potentiellement immense. Les différentes représentations ne sont pas seulement une manière de repérer différents types de rapports entre les espaces de paramètres, mais sont aussi une puissante aide à la pensée, dans le sens où une représentation peut influencer plus ou moins directement le raisonnement. Le musicologue Jean-Marc Chouvel indique qu'une représentation est d'autant plus pertinente qu'elle est prête pour l'interprétation, c'est-à-dire qu'elle résume un réseau de questions musicales. Il appartient au musicologue de valider ou de susciter des représentations, notamment visuelles, qui soient porteuse de sens musical, et donc éligibles à constituer des espaces pour l'imaginaire. De ces nouvelles « visions » de la musique, on peut sans doute espérer de nouvelles idées musicales. Au fond, c'est ce qui est déjà à l'œuvre dans les logiciels musicaux populaires, séquenceurs audio et midi, échantillonneurs, programmeurs de boucles, et c'est ce qui s'expérimente dans les logiciels de recherche conçus pour une utilisation plus savante.

L'informatique musicale comme modélisation de la musique, ou seulement de la logique musicale ? Et qu'est-ce qu'une logique musicale ?

2.2. Logique, non ?

C'est moins avec la logique, au sens habituel du terme, qu'avec les systèmes et les langages formels, que le rapprochement de la musique semble le plus pertinent. En effet, la logique procède par un enchaînement de dérivations et de réductions sur des chaînes symboliques qui finissent par substituer un énoncé terminal à un ensemble d'énoncés intermédiaires. Dans le cas de la musique, l'engendrement temporel ne procède pas véritablement par substitution dans la mesure où ce qui est ex-

primé une fois l'est sans retour, et ne vient pas se substituer, dans un espace purement formel, à une autre expression ; au contraire, la perception d'un antécédent conditionne celle d'un conséquent, et la flèche du temps interdit l'écriture d'un signe d'équivalence entre les termes successifs. Le signe d'équivalence ne pourra alors être utilisé qu'entre des termes abstraits de la structure profonde de la musique, et non pas entre des termes successifs de la surface musicale.

La question du rapprochement entre logique musicale et logique tout court n'a commencé à être discutée sérieusement qu'au XXe siècle. Elle ne pouvait pas être sérieusement posée avant le dépassement des théories primitives de la vérité et la généralisation de la logique aux systèmes et langages formels, c'est-à-dire avant la mathématisation de la logique déclenchée par Boole au XIXe et la logicisation des mathématiques promue notamment par Russel, Whitehead et Hilbert au début du XXe.

En effet il n'y a pas de valeur de vérité en musique et il est vain de chercher dans les structures d'enchaînement musical des structures de raisonnement, pour la raison déjà avancée que la flèche entropique du temps d'écoute interdit le signe d'équivalence entre les termes successifs. Par contre toute l'histoire de la musique confirme qu'il n'est pas absurde d'en considérer le discours, à un certain niveau, comme un langage formel, c'est-à-dire comme un ensemble d'expressions bien formées relativement à un système de règles et d'objets primitifs posés comme axiomes (que certains appellent aussi le matériau).

La mutation du compositeur en bâtisseur de système formel est notamment illustrée par la révolution dodécaphonique et sérielle, dans laquelle les axiomes ne sont pas des objets directement dictés par la perception (ils accèdent alors par leur arbitraire même au statut indiscutable d'axiomes) et les règles de construction s'émancipent du passé. La mutation est menée à un stade proche de la saturation dans la période contemporaine, où ce mécanisme de refondation formelle se voit mis en œuvre avec une granularité de temps qui ne ressort plus de l'échelle historique et se réduit quelquefois à la période de gestation d'une seule œuvre.

Ces deux évolutions de la logique et de la musique vers la notion de système (ou de calcul) formel sont quasiment concomitantes, et éclairent d'un jour singulier la relation de la musique à l'informatique.

Le ordinateur est alors un être tout de langage (formel), mais de langage en action, qui dit ce qu'il fait et fait ce qu'il dit : le stade ultime de la performativité. L'informatique serait une modalité de l'évolution qui rendrait asymptotiquement convergents l'ordre du monde et celui du discours, dans la mesure où elle change aussi l'ordre du monde par son discours. De ce point de vue le philosophe prophétique est Leibniz, avec sa caractéristique universelle, plus que Descartes, car il introduit la dimen-

sion combinatoire du système symbolique, propice aux échappées hors des intuitions premières, donc à la créativité, notamment musicale.

2.3. Computer/ Composer

Qu'est-ce alors que composer avec un ordinateur ? Tout d'abord créer une situation expérimentale à partir d'axiomes (le matériau, les présupposés musicaux) et de règles (les relations qui fondent le système formel), qui s'articulent dans un modèle paramétré. Ensuite c'est observer une simulation du modèle, avec des rendus visuels, auditifs, voire haptiques, sélectionner des éléments intéressants, puis, le cas échéant, reboucler et affiner les axiomes, les règles ou les paramètres.

Par exemple, dans la musique spectrale (un mouvement important de la musique contemporaine né en France dans les années soixante-dix et qui a ensuite diffusé mondialement jusqu'à être enseigné dans des universités américaines, notamment par le compositeur Tristan Murail) l'axiome est constitué par l'analyse de réalités acoustiques (le bruit de la mer, un instrument qui joue, la parole humaine). Le système formel est principalement harmonique : il crée des relations entre les constituants du son obtenus à l'étape précédente (les hauteurs, durées et intensités présentes dans le spectre) qui reflètent dans un premier temps le réel (relations des harmoniques dans le spectre) et s'en échappent dans un deuxième temps par le jeu combinatoire des transformations spectrales (dilatation, compression, transposition, modulations diverses) en créant des systèmes harmoniques inédits (Murail 1982).

Dans une approche qui va au-delà des premières expériences spectrales, c'est la modélisation cognitive de l'auditeur qui est mise à profits. Ce ne sont plus les paramètres objectifs du son, tels qu'ils peuvent être mesurés par des machines (oscilloscope, spectrographe) qui constituent l'axiome, mais les critères subjectifs de perception ou d'appréciation, obtenus par modélisation ou enquête. Ainsi d'une œuvre de Roger Reynolds, *The Angel of Death*, réalisée pour l'Ircam en 2001 collaboration avec des spécialistes de la perception, Stephen McAdams et Emmanuel Bigand, dans laquelle les matériaux musicaux choisis par le compositeur (thèmes, timbres, textures, transformations...) sont agencés en tenant compte des données expérimentales recueillies sur leur perception. par le public (McAdams & Battier 2004).

Chez les tenants de la synthèse du son et de son mariage avec le son naturel des instruments, tels Marco Stroppa ou Mauro Lanza, la dimension formelle de l'écriture est repliée à l'intérieur du matériau sonore lui-même, tout en continuant à se déployer à l'extérieur dans la forme. Le défi informatique est ici de créer une continuité entre cet intérieur et cet extérieur, l'écriture du son et l'écriture des formes. La problématique est celle évoquée plus haut des transductions entre couches de codage qui sont aussi des niveaux de signification dans la musique : le niveau du signal (le son numérique enregistré ou synthétisé) et celui du signe (l'écriture).

Pour les amateurs de temps-réel, le problème est celui du ordinateur-instrument qui réagit au quart de tour en situation de concert et en fonction des autres instruments. Pierre Boulez a inauguré ce paradigme dans son œuvre *Répons*, et Philippe Manoury continue de le raffiner dans la plupart de ses œuvres.

L'improvisation elle-même n'est pas exclue du champ informatique, puisque des expériences d'apprentissage sont menées qui permettent de capter la structure logique d'un flux musical (enregistré ou produit par un instrumentiste) et d'en restituer des variantes, donnant lieu à des situations d'interaction intéressantes.

L'énumération pourrait continuer, incluant l'immense palette des effets audio-numériques utilisés quasi-systématiquement dans toutes les situations musicales, la spatialisation du son musical, et bien d'autres techniques.

Mais le point important reste la convergence de la musique et de l'informatique, deux activités productrices d'êtres de langage, où circulent et se traduisent sans cesse d'un code à un autre des signes performatifs qui font ce qu'ils disent et disent ce qu'ils font.

3. Premiers pas à l'Ircam

Juste après mon DEA Langage Algorithmes Programmation passé en 1985 sous la direction de Pierre Cointe à au laboratoire LITP de l'UPMC j'ai été embauché par l'Ircam comme chercheur junior sous contrat. Je fréquentais cet institut depuis 1983 grâce au compositeur Claudy Malherbe qui m'avait invité à y rejoindre, de manière informelle, l'Atelier de Recherche Instrumentale de Pierre-Yves Artaud. Tout en poursuivant mes études, je contribuais aux travaux de ce département à travers une recherche sur les nouvelles techniques de jeu instrumental et leur modélisation en vue de leur exploitation formalisée dans un cadre compositionnel. Cette étude eut un impact inattendu à l'Ircam, en constituant la première approche scientifique qui permette de jeter un pont entre psycho-acoustique instrumentale, technique instrumentale contemporaine, bases de données sonores, formalisation musicale, et — le terme et ce qu'il sous-tend étant encore à ce moment relativement inédits — la composition assistée par ordinateur (CAO). Cette initiation à marche forcée à la recherche résultait rapidement en un rapport de recherche [Assayag, Castellengo, Malherbe 1985a] et un papier accepté à l'International Computer Music Conference organisée en 1985 à Vancouver [Assayag, Castellengo, Malherbe 1985b].

3.1. Nouvelles techniques instrumentales

Cette étude menée de 1983 à 1985 avec Claudy Malherbe et Michelle Castellengo se déclinait selon trois axes :

- constitution d'une base de données de milliers d'échantillons instrumentaux avec méta-données sur les techniques de jeu sous-jacents
- fouille de données à l'aide de l'algorithme de Terhardt — dont c'était la première utilisation en recherche musicale, et une des premières implémentations effectives proposée par Dan Timis — pour l'extraction de signatures harmoniques. L'algorithme de Terhardt (Terhardt & al 1982) est un algorithme perceptif qui élimine du spectre sonore des composantes masquées, et en ce sens un précurseur du MP3.
- conception d'un environnement interactif pour le compositeur permettant de naviguer dans cette base de connaissances musicales et de construire des parcours harmoniques obéissant à certains critères d'optimisation par fermeture transitive d'un graphe d'intersection construit à partir des modèles spectraux de Terhardt. Cette dernière partie constituait le cœur de mon mémoire de DEA [Assayag 1985].

J'avais bénéficié d'un environnement technique et de rencontres remarquables, notamment Michèle Castellengo, directrice du Laboratoire d'Acoustique Musicale de Paris 6, Pierre Cointe, qui travaillait sur l'environnement de contrôle de la synthèse Formes (Rodet & Cointe 1984), Patrick Greussay qui tenait un séminaire futuriste sur les architectures massivement parallèles, André Riotte, compositeur et pionnier de la formalisation et de l'informatique musicales dont j'avais suivi les cours à Paris VIII.

3.2. L'attente d'un environnement pour la composition

Parallèlement à ce travail sur les techniques instrumentales, j'avais en collaboration avec Claudy Malherbe et André Riotte sur un environnement généralisé pour la formalisation des structures musicales et la composition, dont nous ressentions l'absence et la nécessité. Je collaborais d'autre part avec Dan Timis sur une problématique corollaire, et tout aussi prospective, l'édition informatisée de notation musicale.

L'idée d'un environnement généralisé de composition assistée par ordinateur opérant sur des représentations formalisées de la musique et donnant un accès productif à la notation communément utilisée par les musiciens constituait alors une sorte d'utopie partagée par de nombreux compositeurs ayant pris goût à la technologie à travers des applications plus orientées vers le traitement sonore. De nombreux exemples historiques indiquaient la voie, tout en se limitant en général à des implémentations particulières pour la réalisation d'une œuvre, ou des bibliothèques de fonctions utilitaires aidant à tel ou tel calcul (Xenakis 1981, Mesnage & Riotte 2006), ou bien se référaient à un modèle stylistique fermé les rendant sans intérêt pour des compositeurs qui tentaient de défricher de nouveaux territoires (Ebcioglu 1992). Le

compositeur Tristan Murail avait développé un outil à son usage personnel sur un micro-ordinateur grand-public de l'époque, le Thomson TO7. Si ce dernier avait une fonctionnalité réduite (le calcul harmonique selon une conception spectrale), il était un des premiers à permettre une visualisation en notation musicale (élémentaire) et même une interaction à l'aide d'un stylo optique (Murail 1982).

Sur le plan de la notation musicale aussi, des travaux antérieurs suscitaient l'intérêt : Mockingbird (Maxwell & Ornstein 1984) proposait des transcriptions d'œuvres pour piano jouées sur un clavier ; SCORE de Leyland Smith (Smith 1987), un système à spécifications textuelles pour la gravure de qualité était même utilisé par un certain nombre de graveurs professionnels. Là aussi, ou bien les systèmes étaient complètement fermés, ou bien correspondaient à un style fermé (e.g. la musique romantique pour piano pour Mockingbird) et dans tous les cas n'approchaient pas, même de loin, la complexité inhérente à l'écriture contemporaine, en particulier l'écriture rythmique.

C'est ainsi que, parallèlement à une ligne de recherche sur le son instrumental et son écriture, je commençai à travailler à un projet beaucoup plus ambitieux, visant à constituer un environnement informatique pour la création musicale, un environnement qu'on appellerait aujourd'hui *opportuniste*, dans le jargon des Interfaces Homme Machine. Ces environnements musicaux idéaux n'ont pas de tâche prédéfinie et limitent aussi peu que possible l'imagination et le sens de l'exploration de leurs utilisateurs compositeurs tout en communiquant avec eux dans des codes graphiques ou sonores confortables relativement à leur éducation et leurs pratiques habituelles. Le nom de code était CAO, pour *Composition Assistée par Ordinateur*. Cette idée n'était pas populaire au-delà d'un groupe de compositeurs un peu visionnaires, l'Ircam étant principalement engagé dans le développement d'une station de travail pour le traitement du son en temps-réel, et cette réflexion autour de la CAO pour cette raison resta un temps souterraine.

3.3. MuScript, un langage pour la notation musicale

Durant les années 1985-1986 j'ai initié avec Dan Timis, qui était à la fois un collègue de l'Ircam et un camarade d'université, une réflexion sur une idée susceptible de porter la question de l'édition musicale informatisée au niveau d'ouverture et de flexibilité qui lui manquait pour pouvoir imaginer des applications innovantes. Nous pensions notamment, au delà de l'activité traditionnelle de gravure, à des applications créatives dans lesquelles ce langage écrit de la musique serait le médium entre un compositeur qui formalise sa pensée et un système qui la prolonge par sa puissance combinatoire.

Ce projet nous était inspiré par l'apparition très récente du langage PostScript d'Adobe et de la façon dont il promettait de révolutionner la question de l'impres-

sion. L'idée est simple : si le système actuateur (ici l'imprimante) contient un interprète pour un langage de haut niveau, il en résulte que les données à lui envoyer sont plus simples et concises et que la bande passante entre les deux systèmes en est relativement augmentée. Nous prévoyions par conséquent une augmentation future du niveau général d'expressivité et de souplesse de tous les systèmes à finalité graphique. Or c'était bien-là ce que nous souhaitions pour la notation musicale entendue comme futur médium des systèmes génératifs musicaux.

Cependant, que PostScript dût devenir le langage sous-jacent de la musique écrite, cela semblait déjà évident, mais cela ne suffisait pas. En effet la question de la notation musicale est beaucoup plus compliquée que la typographie textuelle (Read 1979). Elle est bi-dimensionnelle, comporte des signes non-locaux (comme les ligatures, les liaisons d'expression ou les indications d'accelerando ou de crescendo). Enfin, le XXe siècle musical a montré que, la musique occidentale innovant essentiellement à travers sa relation à l'écriture, le code graphique lui-même fait l'objet d'évolutions significatives (tout en gardant généralement une certaine cohérence sémantique ou syntaxique avec les canons classiques). Toutes proportions gardées, les expressions de notation codées en PostScript seraient aussi complexes et lourdes à gérer que l'étaient celles de la typographie textuelle avant PostScript.

Influencé par les travaux récents autour d'ObjVlisp (Briot & Cointe 1986) qui montraient la construction rigoureuse d'un langage à objet à partir d'un substrat Lisp purement fonctionnel je proposai alors de considérer qu'une impression d'un message hautement complexe comme une partition musicale devait procéder en trois temps [Assayag & Timis 1986] [Assayag & Timis 1987] :

- envoi d'un premier programme PostScript très compact codant pour une extension objet de PostScript même (langage *ObjScript*)
- envoi d'un ensemble de définitions de classes en *ObjScript* modélisant les objets typographiques musicaux, les objets de liaison dynamiques, les comportements de déformation associés sous forme de méthodes (framework *MuScript*)
- envoi du programme (en *ObjScript* sur le framework *MuScript*) spécifiant la partition

Une autre originalité de ce travail tient à la conception même de l'exécution du dessin d'une partition. Plutôt qu'une conception géométrique véhiculant de nombreuses données de coordonnées planaires, l'approche était plutôt topologique, les classes et les méthodes *MuScript* permettant de représenter une abstraction de la partition où l'on spécifie en particulier la structure de connectivité des objets musicaux, à l'aide de messages que s'envoient les objets «souhaitant» se connecter. D'autres messages dénotent des déformations d'objets de liaison (ligatures, hampes) , susceptible de déclencher, par propagation, des messages de déplacements locaux assurant la cohérence de la topologie.

Ainsi, pour spécifier un groupe de deux croches ligaturées le code MuScript contient l'instanciation des têtes de notes, des hampes, de la ligature (beam), puis l'envoi des messages d'interconnexion, et éventuellement de placement par rapport à la portée ou de déformation comme l'élongation de hampes. Ces messages peuvent résulter à l'exécution en une propagation maintenant la cohérence topologique, par exemple l'élongation de la ligature consécutive à l'élongation des hampes. Une telle description est purement abstraite et correspond à une infinité de réalisations : les têtes de notes peuvent être sur différentes ligne de la portée, leurs positions verticales peuvent être plus ou moins écartées. En envoyant à ces têtes de notes d'autres messages de connexion pour les placer sur les portées ou des messages de déplacements (exprimés en unités logiques de partition et jamais en coordonnées absolues) les modifications implicites de coordonnées et de taille absolues sont propagées par MuScript.

Ainsi, bien que nous ne disposions pas à ce moment d'environnements graphiques interactifs (mais seulement de mini-ordinateurs Vax/Unix et de terminaux textuels) nous adoptons d'emblée une conception dynamique et interactive du système de notation, en misant sur de futurs interprètes postscript intégrés aux systèmes d'exploitation. Le système était susceptible de subir dans le principe des interactions (déformation et propagation à l'intérieur d'une partition) même si ces dernières ne pouvaient encore être expérimentées dans le cadre d'un éditeur graphique. Mais plus profondément, le formatage de partition était conçu comme un ensemble d'interactions dynamiques opérées sur une abstraction topologique de la partition. Cette approche radicalement nouvelle comportait plusieurs avantages :

- une succession d'interpréteurs ou de machines virtuelles (l'interprète PostScript, la machine ObJScripT, la machine MuScript, le programme de formatage qimplémentant les règles de gravure) augmente progressivement le niveau d'abstraction et l'expressivité des programmes spécifiant une partition musicale
- la décomposition du code de notation en constituants logiques libérés des contingences géométriques et reliés par une topologie de connexions ouvre ce dernier à une expérimentation pour l'invention de nouvelles formes graphiques (par exemple, connecter une liaison d'expression à des têtes de notes comme on le ferait avec des ligatures de rythme, cf. *Klavierstück IX* de Stokhausen)
- la conception innovante du formatage comme algorithme de propagation à partir de déformations locales ouvrant aux concept encore prospectifs à ce moment de *partitions interactives*. En particulier ce travail définit pour la première fois rigoureusement les niveaux graphiques, logiques et sémantiques de la notation musicale, que tout système efficace devrait identifier et séparer, un aspect qui sera repris ensuite par plusieurs travaux sur la notation (Stickney 1987) (Dannenbergh 1993).

Techniquement, ObjScript est un langage à objets obéissant au modèle des prototypes (Briot 1989). Il est implémenté en PostScript en tirant parti du fait que les fonctions dans ce langage sont des objets de première classe et qu'il est par conséquent possible de construire par programme des fonctions PostScript et de les stocker dans les variables d'instance des objets représentant des classes. ObjScript possède les concepts de classe, d'héritage, de méthodes et de variables d'instance, mais il n'a pas de méta-classes. C'est le premier PostScript objet.

Lors d'un voyage en Californie, Dan Timis et moi-même avons tenté de persuader M. Geshke, alors directeur d'Adobe, d'intégrer ObjScript dans leur standard. Ce dernier nous a dit avoir beaucoup aimé le concept, mais a précisé que le marché musical n'était pas une priorité pour Adobe.

J'ai par la suite aiguillé avec Dan Timis la réflexion, initialement orientée vers l'édition et l'impression de partitions, vers la question du stockage, du transfert et du partage de partitions [Assayag & Timis 1987], en mettant l'accent sur la robustesse des représentations de très haut niveau soutenues par des machines virtuelles publiques. Il est regrettable que le marché n'ait pas suivi ce type d'option et multiplié les formats bas niveau et propriétaires : il en résulte indubitablement une grande pagaille jusqu'à aujourd'hui. Mais mon intérêt était vraiment tourné vers les systèmes d'aide à la création et cette question de la standardisation numérique de la notation musicale n'était qu'une étape.

3.4. Crime, un environnement pour la composition

J'ai mis en place ce projet logiciel à l'Ircam en 1985-1986 dans le cadre du groupe de recherche monté par Claudy Malherbe. Il est intéressant de rappeler le contexte. Après avoir mené avec succès le projet de modélisation des nouvelles techniques instrumentales et lancé la recherche sur les représentations du code écrit de la musique, j'ai pu montrer les premiers résultats d'une exploration, encore discrète à l'Ircam, visant à proposer une approche généralisée de la formalisation des structures musicales et une station de travail pour le compositeur. Il en est résulté un rapide intérêt de l'institution pour un angle de vue qui n'y était pas jusque lors représenté — et d'ailleurs très peu étudié ailleurs comparativement à l'effort massif sur le traitement du signal musical. J'ai alors eu la possibilité, bien qu'encore étudiant, de mener ce projet pendant deux années. Le groupe initial avec C. Malherbe, A. Riotte pour l'expertise musicale, D. Timis E. Amiot et moi-même, pour les aspects mathématiques et informatiques s'est alors organisé en une structure légère : la Cellule de Recherche Instruments Modèles Ecriture, ou Crime. J'ai alors mis en chantier l'environnement informatique homonyme en vue de fournir à l'Ircam son premier instrument de Composition Assistée par Ordinateur clairement orienté vers l'écriture

musicale, le logiciel Formes (Rodet & Cointe 1984) étant lui orienté vers le contrôle de processus de synthèse sonore.

J'ai choisi une représentation formelle de la musique susceptible par sa généralité de supporter des opérations d'analyse et de composition en réduisant au maximum les hypothèses stylistiques, condition sine qua non pour mettre en place un environnement de *découverte* [Lartillot 2002a]. Inspirée par Iannis Xenakis (Xenakis 1981) et explorée intensivement à ce moment là sur le plan musical par Malherbe et Riotte, cette représentation dite en *Cribles* a une base arithmétique (les classes de résidus modulo n) et une structure algébrique (l'algèbre ensembliste). Tout complexe musical de hauteurs et de durées peut être analysé en une expression formelle canonique dans cette algèbre et réciproquement toute expression construite dans cette algèbre est susceptible de recevoir une interprétation significative dans le domaine des hauteurs et des durées. Riotte (Mesnage & Riotte 2006) a montré que ce formalisme est doté d'une bonne expressivité musicale dans le sens où les termes d'une expression représentant un complexe hauteurs/durées ont eux-mêmes une interprétation qui met en relief des constituants analytiques musicalement signifiants. Ainsi par exemple d'un rythme complexe dont les constituants révéleront les structures localement périodiques.

L'environnement Crime, écrit en Langage Le_Lisp (Chailloux, Devin & Hullot 1984) de L'Inria a rassemblé un certain nombre de concepts et de techniques qui étaient inédites à l'Ircam et représentait une vision de l'informatique musicale innovante :

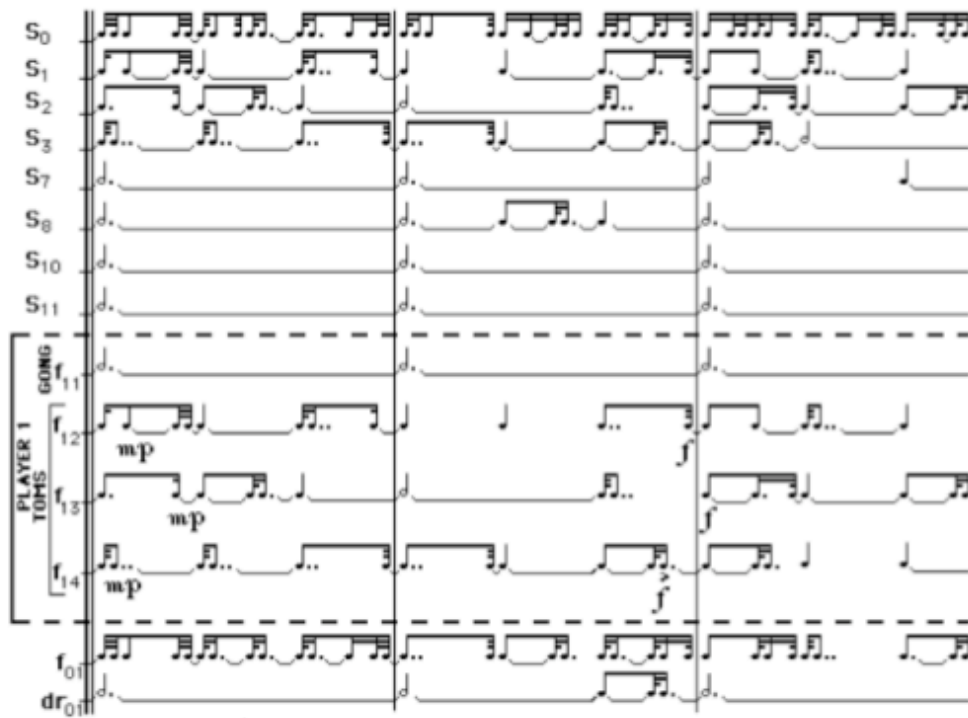
- représentation unifiée des structures musicales temporelles et de leurs contenus en hauteurs à travers un formalisme algébrique simple et universel
- définition d'un interprète écrit en Lisp pour un langage permettant la manipulation interactive des structures musicales ainsi formalisées
- interface d'impression des résultats en notation musicale sans limite de complexité métrico-rythmique ou harmonique [Assayag, Amiot & al 1986]

A l'aide de Crime le compositeur pouvait spécifier des matériaux harmoniques, mélodiques et rythmiques littéraux où résultant d'un calcul, en intervenant à divers niveaux d'abstraction par l'écriture de formules algébriques pour :

- des réservoirs structurés de hauteurs
- des cadres métriques et des figurations rythmiques
- des parcours mélodiques ou des agrégations harmoniques

A partir de ces éléments de matériaux, l'algèbre sous-jacente permettait toute combinaison pour composer des structures de plus haut niveau tout en vérifiant chaque étape de construction par la notation musicale. Cette dernière atteignait alors dans le système un niveau de sophistication inédit en informatique musicale et permettait

de représenter les complexités rythmiques inhérentes à la musique du XXe siècle et donc d'expérimenter librement dans la création de nouvelles formes musicales.



Sortie imprimée de l'environnement CRIME : extrait de l'œuvre algorithmique Partition-gouffre pour percussions d'André Riotte, 1986 (Assayag, Amiot & al 1986)

4. Développements

A partir de 1987, j'ai dû faire une pause dans mes activités de recherche dans le domaine de l'informatique pour l'écriture musicale. En effet je n'avais pas de statut permanent à l'Ircam et suite à une réorganisation et un redéploiement des priorités vers le temps-réel, cette branche, que j'avais contribué à créer avec mes collègues du Crime, et qui avait suscité une grande curiosité, devenait luxueuse pour l'institution. Quelques années plus tard, cet axe de recherche devait devenir un des plus importants de l'Ircam et un de ses atouts maîtres dans la compétition avec les organismes concurrents, mais nous n'y sommes pas encore.

En attendant j'ai principalement mené une activité free-lance en R&D informatique sans quitter le domaine du calcul pour le son et la musique, et d'ailleurs, avec comme client principal l'Ircam et une Start-up issue du même organisme.

J'ai ainsi collaboré avec la société Act-Informatique pour le portage de Le_Lisp de l'Inria sur Apple Macintosh (1987) : j'ai travaillé principalement sur la compilation du langage intermédiaire LLM3 vers le code Motorola 68000 et sur les aspects graphiques et concurrent du langage. Ce langage a été utilisé pour la réalisation du premier environnement fonctionnel pour le contrôle concurrent de processus musicaux utilisant la norme Midi (Boynton & al 1986).

J'ai aussi développé le logiciel NexoCaad, en collaboration avec l'équipe Acoustique des Salles de l'Ircam (sous la direction de Jean-Pascal Jullien) et avec le fabricant d'enceintes de sonorisation Nexo. Ce logiciel contient un modèleur 3D pour concevoir des salles de concert et un modèle statistique de propagation du son pour simuler la distribution des réflexions précoces et tardives et prédire la qualité acoustique de la salle en tout point. Ce logiciel a été utilisé notamment pour la conception de la salle de concert de l'Opéra Bastille à Paris.

Avec la même équipe de l'Ircam et la start-up APIA, j'ai développé AMS (Apia Measurement System) un système de mesure de réponse impulsionnelle permettant de caractériser acoustiquement un espace de concert, à base de PC portable et de carte de traitement de signal dédiée TMS320. Cet programme a permis une campagne de mesures européenne dans les plus grandes salles de concert aux fins de constituer une base de données de modèles pour configurer la salle à acoustique variable de l'Ircam.

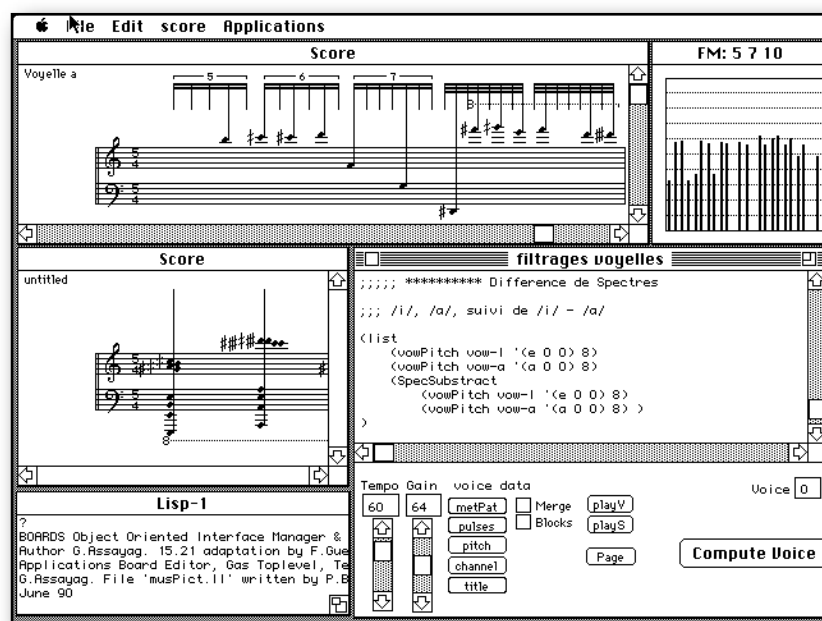
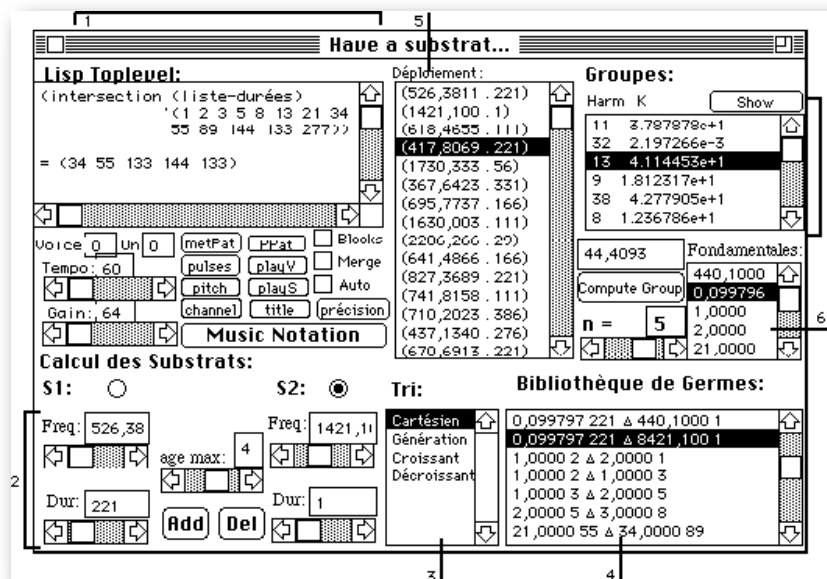
J'ai enfin participé au projet « Le Spatialisateur » de l'équipe Acoustique des salles de l'Ircam avec Olivier Warusfel, Jean-Pascal Jullien et Georges Bloch [Assayag, Bloch, Jullien & al 1992]. Ce système constitue un moteur d'acoustique virtuelle radicalement nouveau puisqu'il permet notamment une configuration par critères psychoacoustiques, ces derniers étant automatiquement réduits à un petit nombre de dimensions porteuses de facteurs objectifs mesurables. Le spatialisateur a été un des tout-premiers systèmes opérationnels de contrôle de l'espace sonore et de la réverbération paramétrable selon des critères perceptifs.

5. Boards

Avant de revenir dans la recherche institutionnelle et en parallèle avec les projets plus orientés ingénierie décrits plus hauts j'ai effectué des recherches personnelles sur un environnement de composition assistée d'un type nouveau (1990).

L'environnement Crime avait été conçu avant l'arrivée massive des ordinateurs à affichage bitmap et interface souris/fenêtres et les sorties graphiques étaient encore destinées à l'impression papier. J'avais d'autre part, à travers ma collaboration avec Act-Informatique autour de Le_Lisp pour Macintosh, rencontré Jean-Marie Hullot à l'Inria. Ce dernier m'avait montré ses environnements Alcyone et SOS Interface

(Hullot 1985) qui devait par la suite, comme on sait, jeter les bases de l'Interface Builder de NexTStep. Le concept de génération d'interface graphiques à partir d'une méta-interface graphique me semblait très excitant. Je décidai alors d'évaluer la pertinence d'un tel paradigme pour la création musicale. Des expériences avaient déjà été menées par Pierre Lavoie (Boynton, Lavoie & al 1986) chez Act-Informatique, mais ces dernières se cantonnaient à la génération d'interfaces à l'aide de déclarations textuelles. Mon idée consistait à donner la possibilité au musicien, sans jamais quitter le niveau graphique intuitif, de spécifier à la fois la logique de ses applications et leurs interfaces spécialisées.



C'est ainsi que j'ai créé l'environnement *Boards* [Assayag 1993], en utilisant le langage Le_Lisp sur MacIntosh, version de ce langage que j'avais contribué à rendre disponible chez Act Informatique. Boards reprend dans Crime l'idée de fournir un langage de programmation algébrique au compositeur, mais il permet d'attacher interactivement des expressions formelles du langage à une interface graphique de telle sorte que l'évaluation des expressions provoque immédiatement un rendu en notation musicale à l'écran. D'autre part, j'ai conçu un système de génération automatique d'interfaces graphiques dans l'esprit du futur Interface Builder de NexStep, dont les atomes étaient des contrôles classiques (boutons, menus, barres de défilement) mais aussi des panneaux pouvant contenir du texte, des graphiques, et en particulier des partitions musicales en conservant évidemment l'ambition de ne pas limiter la complexité des structures représentées (notamment rythmiques). La spécification d'une interface s'effectuait à l'aide d'une interface elle-même graphique permettant d'associer des méthodes aux actions. Bien évidemment le système était écrit par la méthode du bootstrap, un constructeur graphique d'interface élémentaire étant écrit en Lisp, puis enrichi par la manipulation du constructeur lui-même, de telle sorte que Boards est finalement écrit en Boards (implémentation méta-circulaire selon la terminologie d'Emmanuel Saint-James).

Le résultat nouveau en informatique musicale était donc la capacité pour l'utilisateur de configurer des documents musicaux riches à base de *composants actifs* connectés à des formalismes de composition. Cette approche n'est pas sans évoquer le concept de document à composants qui sera proposé dans la technologie *OpenDoc* d'Apple (Apple 1994) quelques années plus tard et censée alors remplacer le concept d'application.

Par rapport à Crime on voit clairement la progression : d'abord un langage textuel, proche d'un langage de programmation, permettant au compositeur de construire des expériences d'écriture musicale. Puis un environnement lui permettant de construire des applications interactives et graphiques spécialisées, et de définir ainsi un cadre d'expérience précis (par exemple les esquisses d'une œuvre donnée) avec une interface adaptée, tout en retrouvant dans les composants actifs le langage formel (l'idée n'étant pas de simplifier les choses au point de perdre la puissance de description).

Boards a été employé notamment par : Claudy Malherbe, *Klang* (1990) pour vents et percussions, *PlayBack* (1991) pour soprano, ensemble instrumental et électronique, *Masques* (1991) pour sept instruments ; Philippe Haïm, *Timeless*, pour treize musiciens (1992).

L'expérience du projet Boards, un projet de recherche personnel mené sans attache institutionnelle sera consignée lors de mon retour à l'Ircam dans un article des Cahiers de l'Ircam [Assayag 1993].

6. Retour à l'Ircam : de PatchWork à OpenMusic

J'ai été recruté comme responsable de l'enseignement de l'informatique musicale aux compositeurs du cursus de composition et informatique musicale de l'Ircam et j'ai exercé cette fonction de 1990 à 1992.

Depuis mon départ de l'Ircam en 1986, le projet CRIME, qui avait laissé une impression durable sur les esprits avait fait des petits. Jean-Baptiste Barrière, responsable du département Recherche Musicale, avait favorisé une série d'expériences qui prolongeaient le concept d'aide formalisée à l'écriture et à la composition que l'équipe du CRIME avait initié. Surtout, les ordinateurs personnels à interface bitmap/souris étaient devenus courants, et permettaient enfin l'interaction directe avec les structures musicales. Parmi les expériences intéressantes qui furent alors mises en œuvres plusieurs méritent d'être rappelées :

- *Pre-Form* (Duthen & al 1990), une tentative d'adaptation du *Formes* (Rodet & Cointe 1984) de Pierre Cointe aux nouvelles architectures à interface graphique
- *Esquisse* (Baisnee & al 1988), un environnement en Lisp, basé sur *Pre-Form* reprenant des principes de composition et du code du compositeur finlandais Magnus Lindberg et du compositeur français Tristan Murail
- *Carla* (Courtot 1990, 1993), une tentative originale de programmation visuelle au-dessus de Prolog pour spécifier des structures musicales de manière relationnelle
- *PatchWork* (Laurson 96) le plus important, en collaboration avec Mikael Laurson, Camilo Rueda, Jacques Duthen. Cet environnement de programmation graphique, pour lequel Tristan Murail a aussi fourni une grosse expertise, jouera un rôle important dans la genèse d'OpenMusic

Plusieurs de ces environnements ont repris certains éléments du code de CRIME, auquel ils succédaient, notamment pour ce qui concerne l'impression de partitions et pour certaines fonctions de recherche musicale. En 1992, *PatchWork* avait gagné la lutte évolutionniste et obtenait un succès certain auprès des compositeurs. Cependant, tous ces projets avaient été réalisés avec peu de moyens, en faisant appel à des bonnes volontés extérieures comme Mikael Laurson, dans la mesure où la revendication de la CAO comme axe prioritaire de recherche et de développement n'avait pas vraiment été assumée par l'Ircam.

Sous l'impulsion d'une équipe dirigeante qui comprenait Laurent Bayle, Jean-Baptiste Barrière, Andrew Gerzso et Jean-Pascal Jullien cette prise de conscience eut lieu en 1992 et la direction de l'Ircam décida de créer une nouvelle équipe de recherche intitulée « Représentations Musicales » et dont la finalité et les objectifs étaient en gros ceux que le groupe informel CRIME avait proposés entre 1983 et 1986.

Laurent Bayle, directeur de l'Ircam me proposa alors de prendre la tête de ce nouveau pôle, doté de moyens à égalité avec les autres équipes de recherche du département R&D et disposant d'une liberté d'initiative significative quant à la définition de ses objectifs et stratégies de recherches.

Je participai alors à la finalisation de PatchWork et commençai de mettre en chantier la réflexion qui devait mener quelques années plus tard à l'apparition d'OpenMusic. Parallèlement à ces grandes décisions architecturales, j'ai défini les stratégies de l'équipe en termes de recherche musicale.

Il est donc utile, arrivé à ce point de nous arrêter un moment pour décrire les stratégies fondatrices de l'équipe, avant de détailler OpenMusic.

7. Fondation de l'équipe Représentations Musicales : stratégies de recherche musicale

En juin 1992 j'ai fondé à la demande de l'Ircam l'équipe Représentation Musicale. Cette dernière avait statut d'équipe de recherche au sein du département Recherche & Développement de l'Ircam, parmi les autres équipes composant ce département, à savoir : analyse-synthèse, système temps-réel, acoustique (des salles et instrumentale), perception et cognition musicales.

La mission de l'équipe était d'une certaine manière de reprendre l'activité de recherche musicale, auparavant confiée au département recherche musicale qui n'existait plus. Le sens de ce basculement dans le département scientifique était clair : il s'agissait de donner un statut scientifique à une activité jusqu'alors discontinuée et mal définie. Deux enjeux majeurs se profilaient : la création future de l'unité mixte de recherche CNRS - Ircam, qui engloberait uniquement les équipes à fort potentiel scientifiques ; la mise au service des compositeurs d'un groupe de recherche et de développement au plus près de la mission de création de l'Ircam. Or ces objectifs à première vue étaient largement inconciliables.

Je faisais donc face à un défi inédit et pour lequel il ne semblait pas exister d'exemples éclairants dans d'autres centres.

Ce serait mentir que d'affirmer que l'organisation conceptuelle et stratégique de nos travaux m'apparut d'un coup. Dans les faits, cette organisation se mit en place progressivement et parallèlement au déroulement du projet OpenMusic, pour se stabiliser à partir de la thèse de doctorat de Carlos Agon que j'ai dirigée. Cette thèse a en effet joué un rôle particulier dans la mesure, où, en plus de fixer la description d'un environnement concret de création qui devait par la suite largement se répandre, elle en livrait le soubassement théorique, notamment sous la forme d'une sémantique

dénotationnelle originale puisqu'exprimée, elle-même, sous la forme d'un langage formel visuel.

Or ce travail pouvait être défendu aussi bien devant la communauté scientifique que devant la communauté musicale. Il montrait que le cahier des charges de l'Ircam dirigé vers la création musicale, et sa relation originale aux sciences, qui devait bientôt se formaliser par la création d'une UMR, étaient compatibles notamment à travers une approche relativement fondamentale de l'informatique.

Bien entendu de telles choses s'étaient déjà produites : les travaux scientifiques de l'Ircam étaient publiés et donnaient lieu à des thèses. Cependant, les outils technologiques dérivés de ces travaux n'étaient pas au cœur de la problématique de création et d'écriture musicale et en constituaient plutôt des accessoires pour la réalisation. Ceci expliquant que l'activité de recherche musicale se situât d'abord en dehors du département scientifique. Pour que s'accomplisse cette révolution, il fallait que l'approche scientifique de la musique s'émancipât de son substrat physique-signal pour s'étendre à l'abstraction algébrique et aux méthodes de l'informatique théorique. C'est ainsi que je procédai pour créer l'identité de l'équipe Représentations Musicales et l'émergence de cette identité coïncida avec celle de l'environnement OpenMusic et avec le travail théorique concomitant.

8. OpenMusic

J'ai lancé en 1994-1995 le grand chantier d'OpenMusic [Assayag & al 1999]. Il s'agissait alors, dans le cadre d'une équipe récemment fondée, de faire le bilan d'une décennie de recherche musicale, de réflexion et de développement sur les architectures des environnements de CAO, et de proposer une évolution décisive susceptible de fixer le standard pour les années à venir. Ce bouleversement était notamment justifié par la constatation que PatchWork, qui avait été une réussite dans son adoption par les compositeurs, était bloqué dans son évolution par certaines limitations fondamentales. En particulier, l'interface de programmation visuelle de PatchWork constituée d'une sur-couche de Lisp posait problème : elle ne donnait pas accès à l'intégralité de la programmation fonctionnelle du langage sous-jacent. Une raison principale pour cela était l'absence de *fonctions d'ordre supérieur* et de mécanismes associés comme la *curryfication* (Barendregt 1984). De plus, bien qu'elle exploitât le langage objet CLOS sous-jacent (Paepcke 1993), cette interface ne donnait pas accès à la programmation objet. Une raison principale pour cela était le *manque de réflexivité* dans le langage visuel (Briot 1989), qui aurait permis de définir une contrepartie visuelle à CLOS.

Une originalité de PatchWork était la possibilité d'associer un éditeur graphique à une fonction. Cette dernière construisait alors un objet CLOS qui constituait en

quelque sortes le modèle dans un schéma MVC (Reenskaug 1979). Cependant, cela était traité de manière ad-hoc pour chaque type d'éditeur, au lieu que les primitives du langage visuel ne permettassent d'instancier de manière standard des objets CLOS (soient associées à des méthodes d'instanciation).

De manière générale, il n'y avait pas de syntaxe et de sémantique formellement définie, ce qui grevait la sûreté du langage et ses évolutions futures.

Après avoir identifié ces verrous j'ai mis en chantier le projet OpenMusic qui consistait pour une large part à *mettre en accord une démarche de recherche musicale et une démarche d'informatique théorique bien fondée*. Mon analyse, qui s'est avérée correcte par la suite, était qu'il n'y avait pas d'autre façon de poser des bases saines pour un *environnement de créativité* qui devrait par la suite supporter des évolutions importantes en réponses aux problématiques relativement imprévisibles injectées par les compositeurs et les musicologues. J'ai eu à ce moment-là la chance de pouvoir proposer cette thématique comme sujet de thèse à Carlos Augusto Agon, qui avait effectué son stage de DEA dans l'équipe [Agon 1994]. Carlos Agon a effectué un travail remarquable à la fois pratique et théorique et a joué un rôle décisif à mes côtés dans la réalisation d'OpenMusic.

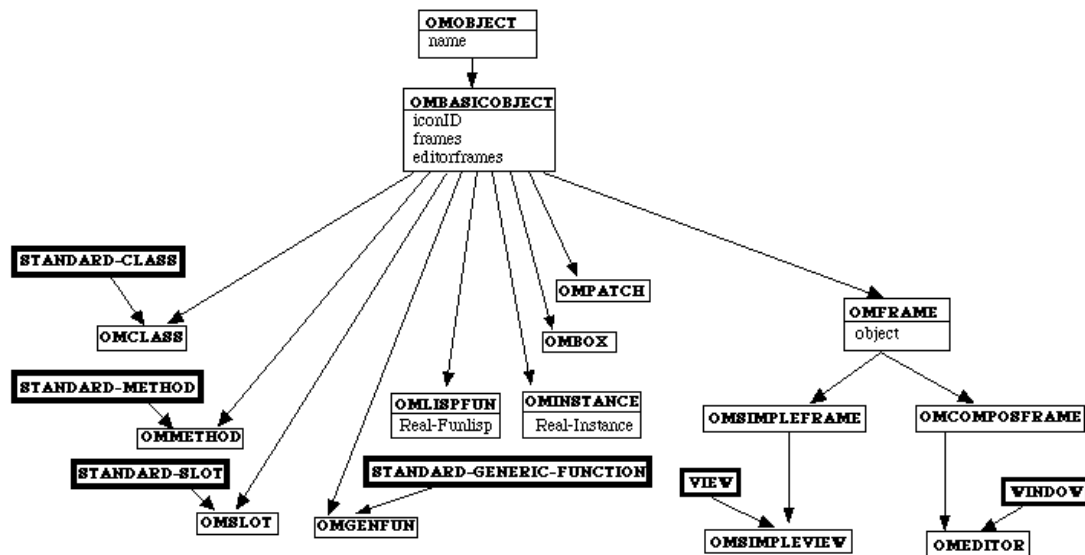
Une innovation principale d'OM est l'adaptation pour la CAO de divers styles de programmation : par objets, fonctionnelle, par contraintes et par méta-programmation. Un autre apport d'OM est la conception et l'implémentation des structures de contrôle graphiques parmi lesquelles les boucles, les conditionnelles, l'abstraction ou la récursion. Cette section montre l'architecture d'OM et ses principales contributions.

OpenMusic est implémenté en CommonLisp/CLOS (Steele 1998) sur Apple Macintosh. Le choix du langage obéit principalement aux raisons suivantes :

- c'est un langage fonctionnel et à objets puissant, bien défini, disposant d'un modèle formel solide (Abadi & Cardelli 1994)
- CLOS et CommonLisp ont fait l'objet d'une normalisation internationale et ont un excellent degré de portabilité (des bibliothèques graphiques orientées objet existent sur toutes les plate-formes)
- il existe une énorme quantité de savoir-faire en informatique musicale associée à ce langage
- le protocole de méta-objets (Paepcke 1993) facilite le type d'extension souhaitée

Le MOP de CLOS contient une partie statique formée d'une hiérarchie de classes de méta-objets et une partie dynamique ou protocole de méthodes, qui permet la manipulation des méta-objets. Nous entendons par méta-objets les éléments du système objet sous-jacent à CLOS, telles que les fonctions génériques, les méthodes, les variables d'instances (slots) et les classes. Nous avons procédé à la définition de sous-

classes des classes de méta-objets et la définition ou redéfinition de méthodes pour ces nouvelles classes ou celles déjà existantes. OpenMusic constitue donc un enrichissement de la syntaxe et de la sémantique de CLOS. En outre, parce que certaines entités, telles que le patch ou la boîte, n'ont pas de correspondant dans CLOS (en effet, il n'y a pas de classe permettant de réifier la notion de programme ou d'invocation fonctionnelle dans CLOS), il nous a fallu définir des nouveaux méta-objets propres à OM. L'un des problèmes décisifs de cette implémentation a été le niveau de détail auquel on définit le protocole : en général, un protocole très détaillé est peu modulaire. Ainsi, les changements doivent être effectués en plusieurs endroits, ce qui affecte la fiabilité. En revanche, un protocole peu détaillé rend délicat le choix de l'endroit où introduire les modifications.



Hiérarchie de classes de méta-objets dans la partie statique d'OpenMusic. Les classes de méta-objets CLOS sont représentées par un rectangle gras.

La hiérarchie de classes dans la figure précédente rend compte de tous les objets manipulables dans l'environnement visuel et dans le langage de programmation visuel d'OpenMusic. Les classes fondamentales du système objet visuel d'OpenMusic héritent d'une part de la classe **OMObject** et d'autre part d'une classe fondamentale CLOS homologue. Les classes **OMPatch** et **OMBox** réifient respectivement les notions de patch (programme ou algorithme) et de boîte (appel fonctionnel) inexistantes dans CLOS. Les classes mentionnées, constituent les objets de calcul de notre langage (partie gauche de l'arbre d'héritage). La partie droite de l'arbre d'héritage implémente le paradigme visuel. On peut noter que la classe **OMFrame** hérite aussi de la classe **OMBasicObject**. Cela signifie que le *paradigme visuel fait partie du langage* : ici se trouve concentrée la *réflexivité* d'OpenMusic, évolution radicale par rapport aux langages de CAO explorés jusque là. Il est important de noter que cette combi-

raison de méta-programmation et de réflexivité, qui permet donc d'*étendre le langage visuel OpenMusic par une programmation en OpenMusic*, a constitué une première, non seulement du point de vue de la recherche musicale, mais aussi de la recherche sur les langages visuels en général [Assayag 1995] [Agon & Assayag 2002] [Agon & Assayag 2003b] [Agon, Bresson & Assayag 2008]

8.1. Protocole dynamique

Les méta-objets dans OM sont composés par un ensemble d'éléments et optionnellement par des relations entre ces éléments. Par exemple, une classe est une liste ordonnée de champs, une fonction générique consiste en un ensemble de méthodes, un patch contient une liste de boîtes connectées, etc.

Le mécanisme de glisser-déposer est le principal outil d'édition dans OpenMusic (comme dans l'environnement de composition ELody (Orlarey & al 1997) de Yann Orlarey, conçu dans la même période). Une opération de glisser-déposer est définie comme une action entre une instance de la classe OMsimpleView appelée "source" et une instance de la classe OMEditor appelée « cible ». Il existe un ensemble de prédicats allow-drop qui déterminent si l'opération de glisser-déposer est possible entre une paire (source, cible). Ce mécanisme est au cœur de la dimension intuitive et expérimentale pour l'utilisateur : en étendant les domaines de ces prédicats, des opérations de glisser-déposer deviennent possible entre à peu près tout et n'importe quoi, à condition de définir la sémantique associée, ce qui se fait en surchargeant une fonction générique qui définit l'ajout d'élément dans un méta-objet. L'utilisateur *explore et découvre* l'expressivité du langage par une seule opération généralisée qui se trouve par ailleurs être au cœur des systèmes d'exploitation à interfaces graphiques.

Une formalisation graphique de la syntaxe et la sémantique de OM est donnée dans la thèse de Carlos Agon [Agon 1998].

8.2. Programmation fonctionnelle

La notion de base dans OpenMusic est le patch. Le patch réifie le concept *d'algorithme*. Les patches contiennent des boîtes (icônes) et des connexions. Les boîtes représentent des appels fonctionnels, tandis que le graphe de connexion représente la composition fonctionnelle. L'évaluation d'une boîte engendre une chaîne d'évaluations correspondant à l'exécution d'un programme.

Le mécanisme *d'abstraction* d'un patch se fait en ajoutant des boîtes d'entrée et de sortie, représentées par des icônes de flèches. Les patches sont des objets de première classe : *une abstraction peut notamment être une fonctionnelle*.

8.3. Programmation par objets

OpenMusic offre une bibliothèque de classes pour la représentation et manipulation d'entités musicales. L'utilisateur peut étendre cette bibliothèque en créant de nouvelles classes, éventuellement en utilisant le mécanisme d'héritage. Il est aussi possible de créer de nouvelles fonctions génériques ou d'ajouter de nouvelles méthodes. Une fois la nouvelle classe définie, on pourra créer par glisser-déposer de cette classe sur une fenêtre de patch, une boîte *factory*. Cette factory constitue alors un *prototype* pour créer des instances de l'objet musical. Une factory est en général associée à un éditeur.

8.4. Programmation par contraintes

Nous avons étendu les éditeurs d'OpenMusic par l'introduction de mécanisme de programmation par contraintes. Les objets OM constituent alors des domaines ou des instanciations pour un CSP. Les prédicats sont exprimés de manière fonctionnelle.

OpenMusic permet de résoudre un CSP en choisissant parmi plusieurs moteurs de contraintes. Un module de génération de code se charge de traduire le patch graphique en un CSP adapté au mécanisme de résolution choisi. Cette traduction n'a aucun effet sur la représentation graphique du CSP, et est de ce fait transparente pour l'utilisateur. Il existe trois moteurs des contraintes dans OM : Situation, Screamer ou OMClouds.

Situation est un moteur de contraintes par forward checking, conçu par Camilo Rueda [Rueda & Bonnet 1998], et muni d'une interface graphique pour OM. L'une des originalités de Situation est la représentation par objets. Un objet est un ensemble de points et de distances. Il permet d'exprimer à la fois des rythmes ou des accords. Les contraintes portent sur les distances internes (entre points d'un même objet) ou externes (entre des points appartenant à des objets différents).

Screamer (Siskind & McAllester 1993) est un système implémenté par Mark Siskind qui ajoute à Common Lisp une forme d'indéterminisme via deux constructions, *either* et *fail*, introduisant des points de choix dans le langage. L'évaluation de l'expression (either e1...en) évalue d'abord e1, puis si l'évaluation de e1 donne un fail, e2, etc.

OMClouds [Truchet, Assayag & Codognet 2003] est une librairie OM, qui résulte de la thèse de Charlotte Truchet [Truchet 2004] que j'ai co-dirigée avec Philippe Codognet. Elle permet de modéliser et résoudre visuellement des problèmes de contraintes. OMClouds utilise un algorithme de recherche locale, la recherche adaptative (Codognet & Diaz 2001), qui s'avère bien adapté à une utilisation musicale grâce à la possibilité d'édition des résultats partiels ou approchés pendant la résolution [Tru-

chet & Codognet 2004] [Truchet, Agon & Assayag 2001] [Truchet, Assayag, Codognet 2001] [Truchet, Diaz, Codognet 2003].

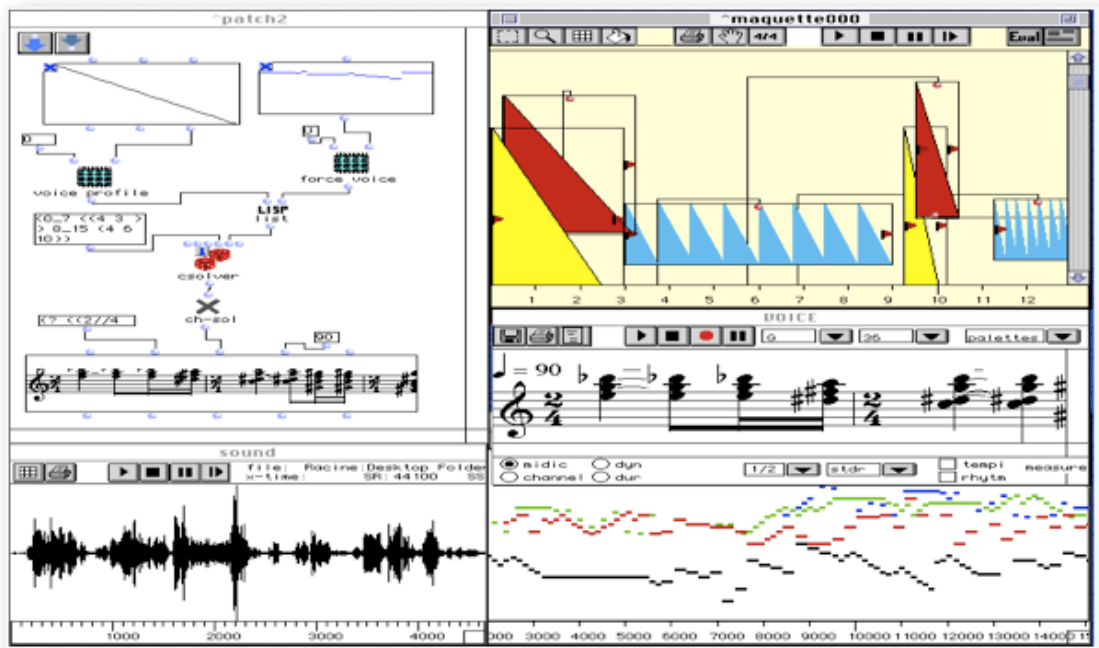
8.5. MOP Graphique

Nous avons utilisé le MOP (Meta Object Protocol) de CLOS pour l'implémentation d'OM. De la même manière que nous avons étendu CLOS lui-même en utilisant des techniques de méta-programmation, l'utilisateur peut étendre OM de manière graphique grâce au MOP visuel [Agon & Assayag 2003b]. Un exemple classique est la réalisation d'une trace d'exécution visuelle en surchargeant la fonction générique *value* qui évalue un patch. Le MOP visuel est utilisé notamment pour l'implémentation des *maquettes*.

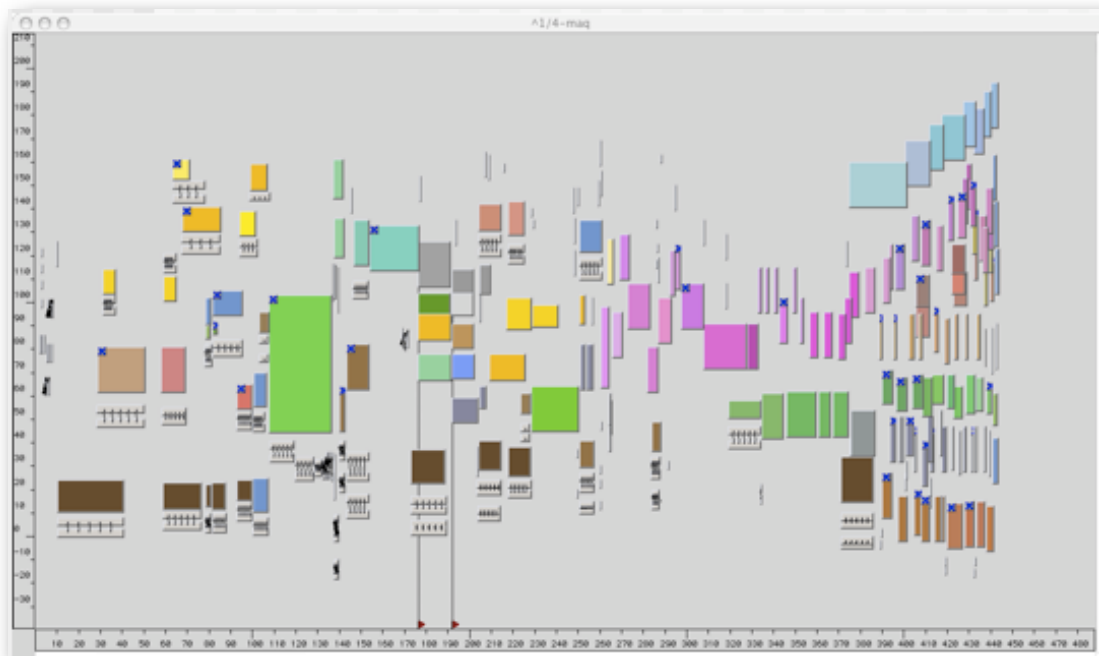
8.6. Maquettes

Les maquettes peuvent être définies simplement comme des « patches dans le temps ». Elles constituent la représentation explicite du temps par laquelle les calculs de matériaux peuvent devenir des esquisses musicales.

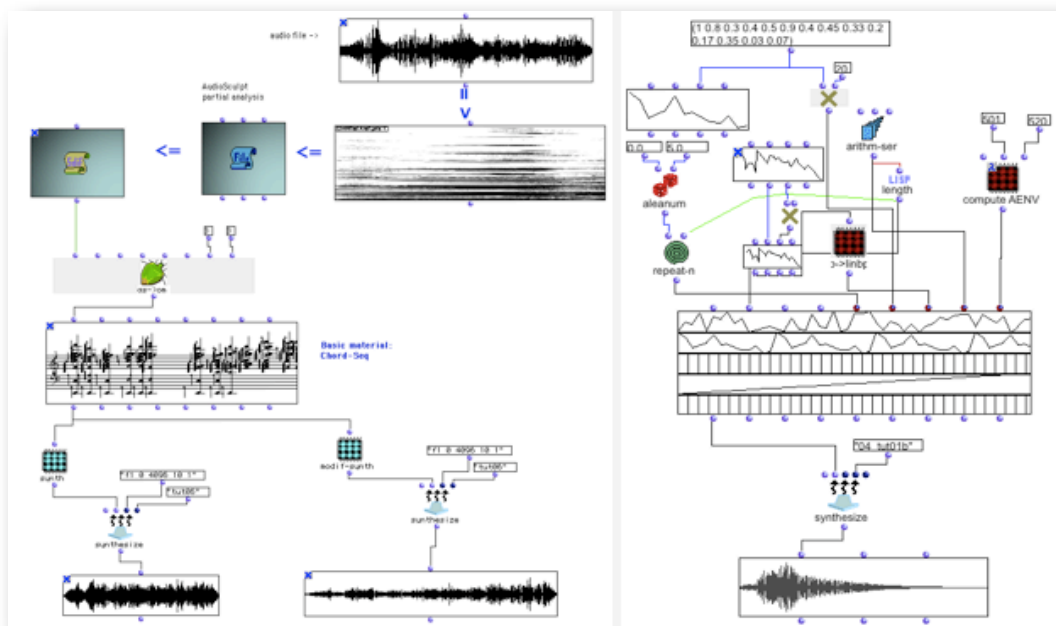
Les maquettes proposent deux relations principales entre les boîtes : une relation de causalité et une relation temporelle. La causalité entre deux boîtes est définie par rapport au « temps d'évaluation ». L'évaluation d'une maquette suit le même paradigme que celui des patches. Ce type d'évaluation étant très simple (nous n'avons pas d'évaluations en parallèle), il permet de définir entre les boîtes une relation d'ordre strict. De même, la règle temporelle de la maquette permet de comparer les objets temporels par rapport à leur « temps physique ». Il résulte de cela que *le temps peut conditionner le calcul* ou bien que *le calcul peut conditionner le temps*, selon la sémantique d'évaluation choisie (grâce au MOP). Dans un cas, la position temporelle est un paramètre, dans l'autre le calcul engendre une position temporelle (et la boîte concernée se déplace ou se déforme alors automatiquement).



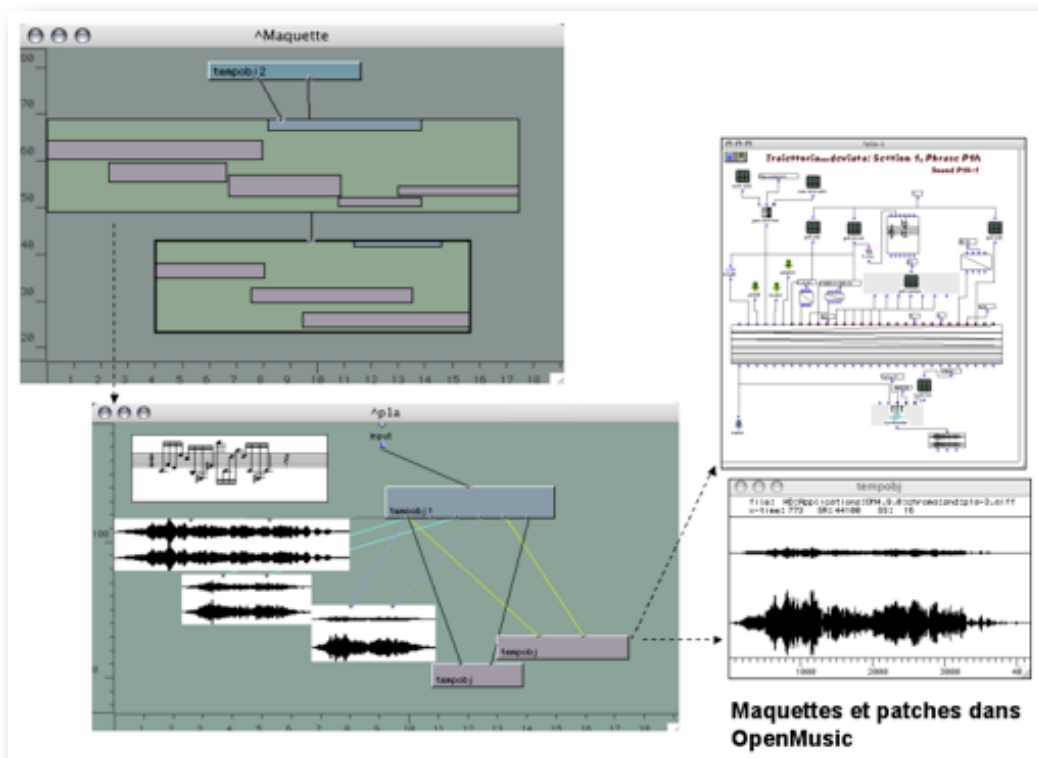
Un écran typique d'OpenMusic : les objets musicaux peuvent être approchés par une variété de représentations



Une maquette du compositeur Tristan Murail.



Aller-retour entre les représentations signal et les représentations symboliques médiatisées par une interface abstraite pour la synthèse sonore



Une Maquette est une mise en temps d'une collection de patches, permettant de représenter et manipuler le temps selon trois modalités : chronologique, causale, hiérarchique.

8.7. Temps et calcul dans OpenMusic

Nous avons donné jusque là une description plutôt informatique d'OpenMusic. Il est intéressant cependant de livrer quelques considérations conceptuelles permettant de croiser les concepts informatiques avec ceux relevant plutôt de la recherche musicale.

Premièrement, nous avons donné un tour résolument fonctionnel à OM : finalement tout est calcul, et l'unité est le programme. Lorsque l'on met en temps un objet musical, ce qui techniquement se fait en glissant un patch dans une maquette, ce n'est pas la donnée musicale que l'on traite, mais le programme conduisant à cette donnée.

Cette caractéristique originale conduit à une grande dynamicité des structures musicales représentées puisque ces dernières véhiculent des constructeurs plutôt que des construits, ces constructeurs étant toujours susceptibles de subir des transformations ou de conférer des sémantiques particulières à des descriptions formelles de structures musicales

La généralisation du glisser-déposer va bien plus loin qu'une simple facilité graphique, elle donne accès intuitivement à une problématique profonde de la recherche musicale : la dialectique hors-temps / en-temps, et cela en résonance avec la dialectique calcul / donnée, plus proprement informatique.

Par cette dialectique, un objet-calcul (un patch) délivre des objets-signes (les instances d'objets musicaux associés à une partition) les deux étant des entités hors-temps (même s'ils évoquent un temps interne comme toute partition). Ces entités hors-temps peuvent être *plongées* (par glisser-déposer) dans un référentiel temporel (la Maquette) et contribuent à former une structure en-temps. Une Maquette, objet en-temps, peut à son tour être *plongée* dans un patch, devenant de ce fait un objet hors-temps, doté de son temps interne. La Maquette devient alors nœud d'un calcul plus complexe. Ce rebouclage est unique à OpenMusic et produit une grande expressivité du langage.

Le couple de dichotomies hors-temps / en-temps et calcul / donnée fonctionne grâce aux aspects fonctionnels d'OM, notamment les fonctions d'ordre supérieur et la possibilité de traiter ces dernières comme citoyens de première classe. Ainsi tout se ramène à une fonction, y compris les structures musicales les plus feuilletées et les plus complexes.

Les maquettes d'OpenMusic permettent de séparer les structures temporelles logiques, chronologiques, causales, hiérarchiques. Elles permettent d'autre part de séparer la structure temporelle de la sémantique. Les différentes structures temporelles sont :

- structures temporelles logiques : relations de logique temporelle (précédence, overlap etc.) (Bestougeff & Ligozat 1989)

- structures temporelles chronologiques : assignation de coordonnées temporelles en millisecondes
- structures causales : connections entre objets temporels au sein d'une maquette. Ces connections modélisent la composition fonctionnelle comme dans les patches, mais sur des fonctions *temporellement situées*
- structures hiérarchiques : les maquettes, récursivement définies, contiennent des objets temporels terminaux (des patches) et des maquettes

La sémantique affectée à une maquette n'est rien d'autre qu'une fonction d'ordre supérieur qui s'exécute sur les composants de la maquette, qu'elle voit comme les composants fonctionnels d'un programme. Par exemple une même maquette, avec une structure temporelle donnée, peut être *évaluée* successivement comme séquence de sons synthétiques avec un modèle de modulation de fréquence, ou bien avec un modèle de synthèse additive, ou bien comme forme purement symbolique exprimée comme partition musicale, etc. La sémantique choisie résulte, comme c'est l'habitude, par un glisser-déposer d'un patch dans une région spéciale de la maquette.

De tout cela ressort la vision originale de la recherche musicale assistée par ordinateur que j'ai voulu promouvoir avec le projet OpenMusic : un environnement où les constructions musicales sont modélisées par des calculs ; où des opérateurs universels, tel le glisser-déposer, permettent de composer naturellement des calculs de plus en plus complexes donc des formes musicales de plus en plus complexes ; où cette complexité est contrôlée de bout en bout par l'association standardisée d'un éditeur à un calcul, ce qui permet de bout en bout de *projeter les états de calcul dans des partitions* ; où des structures temporelles non triviales peuvent coexister avec une sémantique musicale variable.

8.8. OpenMusic et Java : jOpenMusic

L'objet de ce stage de DESS (Génie des Logiciels Applicatifs – UPMC), réalisé par Florent Selva avec Carlos Agon, était de proposer une solution Java (notamment pour des raisons de portabilité et de maintenance), se substituant à la partie visuelle d'OpenMusic. Nous avons réalisé un prototype graphique pour le « kernel » d'OpenMusic. Ce prototype est implémenté par une API CLOS encapsulée en Java sans modification de l'architecture d'OM et une architecture client serveur (client Java, serveur Lisp). Cependant un problème majeur a été repéré : l'impossibilité de délégation graphique pour des objets composites. Ce stage a été le premier pas de réflexion pour un portage d'OpenMusic dans d'autres plates-formes et d'autres langages.

8.9. Applications pédagogiques d'OpenMusic

MusiqueLab 2 est un projet initié en 2004 à l'Ircam en partenariat avec le Ministère de l'Education Nationale et le Ministère de la Culture et de la Communication, visant à développer des applications pédagogiques destinées à l'enseignement de la musique. L'équipe Représentations Musicales a pris en charge le développement d'une application innovante appelée ML-Maquette, dédié à la manipulation et au traitement du matériau musical symbolique, et basé sur le logiciel OpenMusic [Bresson, Guédy, Assayag 2006]. Carlos Agon et Jean Bresson se sont fortement impliqués dans le développement et Fabrice Guédy a été le coordinateur pédagogique pour l'ensemble du projet.

ML-Maquette est un environnement dans lequel un utilisateur, professeur ou élève d'une classe de musique, peut créer et transformer des objets musicaux (notes, accords, séquences d'accords, etc.) Ces objets peuvent être manipulés à l'aide d'éditeurs graphiques, et transformés par divers opérateurs applicables par des processus simples de glisser-déposer : un objet déposé sur un opérateur provoque l'ouverture d'une fenêtre permettant à l'utilisateur d'entrer les différents paramètres d'une transformation à effectuer, puis la création visuelle d'un nouvel objet. Par la suite, les objets peuvent être disposés dans une « maquette » spéciale, dans laquelle ils sont ordonnancés temporellement. La position d'un objet dans cette maquette peut, le cas échéant, modifier des paramètres ayant servi à la création de l'objet et le recalculer en conséquence.

ML-Maquette permet ainsi de constituer des documents pédagogiques abordant des formalismes et concepts musicaux divers tels que la manipulation des différents paramètres des structures musicales (hauteurs, rythmes, dynamique), les transformations et opérations sur ces structures, l'harmonie, le contrepoint, l'intégration temporelle des objets musicaux. Il s'agit finalement, avec cet environnement, de concrétiser des situations pédagogiques dans lesquelles sont étudiés des concepts ou extraits musicaux, dans une interaction pédagogique entre professeur et élève.

Les retours d'utilisation « grandeur nature » dans des classes de musiques du secondaire, ou des stages de formation organisés à l'Ircam [Assayag, Bresson 2006], [Guédy, Bresson, Assayag, 2007] ont grandement aidé à finaliser ML-Maquette.

Une série de nouveaux opérateurs a également été développée, liée aux nouvelles possibilités d'analyse et de synthèse du signal sonore disponibles dans OpenMusic. Ces opérateurs permettent notamment un travail sur les timbres sonores (représentation additive) et sur les relations entre le son et les structures musicales symboliques. Ces nouvelles fonctionnalités se positionnent de façon complémentaire avec les outils de traitement du signal en temps réel proposés dans ML-Audio, une autre application développée dans le cadre de Musique Lab 2 pour le traitement et l'interaction sur le signal sonore.

9. Un support expérimental pour la recherche musicale

Une des fonctions majeures d'OpenMusic à l'Ircam a été de servir de support expérimental aux travaux de recherches des membres de l'équipe et notamment des étudiants accueillis en stage de DEA puis de Master et aux thésards. Le terme expérimental doit être explicité s'agissant de recherche musicale. Une des hypothèses qui sous-tendent tout mon travail est que l'accès aux ressources computationnelles permet aux musiciens d'adopter une démarche expérimentale comparable à celle qui est en vigueur dans les sciences. La création à l'aide de l'ordinateur devient alors un processus cyclique hypothèses - modélisation - simulations - validation. Les sorties du système à chaque étape constituent des matériaux qui seront incorporés dans les œuvres. Ce processus s'applique bien évidemment à l'analyse c'est pour cela que les travaux en musicologie computationnelle [Lartillot-Nakamura 2004] [Lartillot & Assayag 2006] (Leman 1997) ont tendance depuis quelques années à le disputer aux travaux de création s'agissant de la communauté qui s'est formée autour d'OpenMusic.

Il est donc temps ici de préciser ces travaux de recherche musicale et de recherche informatique, qui font l'objet des sections suivantes. Les tous premiers travaux ont été réalisés autour de PatchWork, rapidement suivis par des recherches autour d'OpenMusic.

10. Modélisation d'œuvres et de systèmes

10.1. Modèle informatique d'une pièce pour clarinette d'Igor Stravinsky.

Lionel Rokita, DEA ATIAM, Ircam - Université Paris VI, 1994.

Dans le cadre de son stage de DEA ATIAM, Lionel Rokita a produit une analyse informatisée de la première des trois pièces pour clarinette solo de Stravinsky [Rokita 1994]. Il a utilisé à cet effet le logiciel d'aide à l'analyse musicale « Morphoscope » de Marcel Mesnage et a bénéficié de la direction musicale d'André Riotte. Le modèle final a été réalisé en PatchWork : à partir d'une série de modules programmés pour la circonstance et qui représentent les opérations combinatoires de base décelées dans la partition, le programme PatchWork reconstitue la partition originale. La technique est donc d'analyse-synthèse comme c'était le cas de l'analyse de Ligeti par Marc Chemillier. (Chemillier 1995) reprise en PatchWork dans [Assayag 1993].

10.2. Programmation par contraintes et harmonie tonale

Jean-Louis Guénégou, DEA ATIAM, Ircam - Université Paris VI, 1998.

La programmation par contraintes est une technique de programmation de plus en plus utilisée en informatique, et plus particulièrement en intelligence artificielle. Sa technique ne réside pas dans l'élaboration d'un algorithme de calcul de solutions construit à partir des données d'un problème, comme c'est souvent le cas dans la programmation classique, mais dans une déclaration de prédicats appelés contraintes qui sont fournis à un moteur de recherche censé explorer un domaine prédéfini de possibilités pour extraire celles qui satisfont aux contraintes. Ces possibilités extraites sont les solutions du problème .

Ce type de programmation s'adapte bien dans le domaine de la composition assistée par ordinateur (CAO), car les problèmes que les compositeurs soumettent à la machine, souvent très intuitifs pour l'homme, sont en fait souvent difficiles et longs à traduire en algorithmes de calculs. De plus, le compositeur est amené très fréquemment à changer, de peu ou de beaucoup les problèmes posés, pour des raisons d'échecs où d'idées nouvelles ce qui justifie une approche déclarative.

Les méthodes visant à améliorer le moteur de satisfaction de contraintes sont souvent des compromis entre l'efficacité de la recherche dans le domaine et la variété de contraintes qui peuvent être supportées. Les améliorations idéales sont bien sûr celles qui rendent la recherche plus efficace sans restreindre le type de contraintes acceptées.

Situation [Rueda, Bonnet 1998] étant un moteur de recherche conçu pour la CAO sur un modèle intervallique, il a été réalisé de manière à pouvoir s'appliquer principalement sur la génération de séquences harmoniques. *Situation* a jusqu'à présent été utilisé pour créer des séquences harmoniques de musique contemporaine, donc utilisé dans des systèmes propres à chaque compositeur. Dans le but d'améliorer et de valider universellement le moteur de *Situation*, on a pensé adopter la démarche qui consiste à tester le moteur sur des exemples appartenant à un domaine connu de tous et pour lequel il existe un corpus de règles bien définies.

Ainsi, on a cherché à générer des séquences d'harmonie tonale. En effet, les règles de l'harmonie tonale mettent en jeu des contraintes de type très divers. De plus, la structure tonale bénéficie d'une évolution musicale à travers plusieurs siècles ce qui en fait un exemple d'application des contraintes pertinent. La méthode généralement appliquée dans la recherche en informatique musicale jusqu'ici a généralement été de faire une modélisation mathématique de règles d'harmonie et de contrepoint. Certains chercheurs ont déjà mené des travaux dans ce sens, c'est le cas par exemple de Kamal Ebcioglu (Ebcioglu 1992) avec des simulations de chorals dans le style de J.S BACH, de David Cope avec un système de base de donnée permettant de générer

des séquences musicales (Cope 1996), de Hornel (Hornel & Ragg 1996) avec un système de réseaux neuronaux capable d'harmoniser dans le style baroque et enfin de P. Barbaud (Barbaud 1968) avec des applications des automates finis en génération d'harmonie tonale.

Spécialement pour le cas de Situation, on s'est proposé de recenser les méthodes de l'harmoniste professionnel pour harmoniser à vue, par exemple le fait de penser à harmoniser d'abord la basse d'un choral, et ensuite les autres parties. Ensuite, on a essayé d'implémenter ces méthodes sous formes de contraintes dans Situation. On essaie de concilier simultanément l'aspect contrapuntique et harmonique, chose rarement faite auparavant.

Les essais sur la musique tonale nous ont donné des idées d'améliorations ou de compromis pour Situation dont quelques uns sont donnés ici à titre d'exemple.

- Le système de notation du domaine étant très important pour une expression simple des contraintes et donc, une recherche efficace, Situation devrait permettre facilement au programmeur d'accepter toute les notations. La notation du domaine apporte déjà une information sur les propriétés. Dans le cadre de l'harmonie à 4 voix, cette notation apportait au système une connaissance de la structure des accords.
- L'ordonnancement dynamique devrait permettre de trouver plus rapidement une solution, même si le domaine est parcouru moins vite. Il faudrait donc ajouter au moteur cette possibilité.
- Il fallait que Situation permette facilement à l'utilisateur d'exprimer dans quel ordre il veut faire l'instanciation des accords lors du Backtracking, car une recherche rapide impose souvent d'instancier d'abord les variables qui ont un domaine très affectés par les contraintes. On sait que les harmonistes à vues prévoient toujours un peu en avance les cadences sur lesquelles ils vont aboutir lors de l'harmonisation de chorals.

10.3. Imagerie musicale pour l'analyse harmonique : OMiel

Benoît Mathieu. DEA MIASH, ENST Bretagne, 2002.

Ce projet a été réalisé grâce à la dynamique impulsée par le groupe de travail « Outils pour l'analyse » que j'ai dirigé 2001 avec Xavier rodet. Il a permis de traiter, en quelque sorte en « temps-réel », des difficultés d'analyse soulevées par les participants au groupe (notamment J.P. Cholleton) et de mettre en œuvre des collaborations directes avec les musicologues présents (en particulier J.M. Chouvel). Le corpus est issu des compositions du compositeur brésilien Hermeto Pascual, dont l'harmonie a pour caractéristique de se situer aux confins de la tonalité et du Jazz, sans toutefois relever clairement de ces modèles : d'où le besoin d'outils nouveaux.

Le projet a comporté deux volets :

- Une proposition pour la classification des accords
- Une proposition pour un outil innovant d'imagerie musicale

Le principe de classification des accords a impliqué :

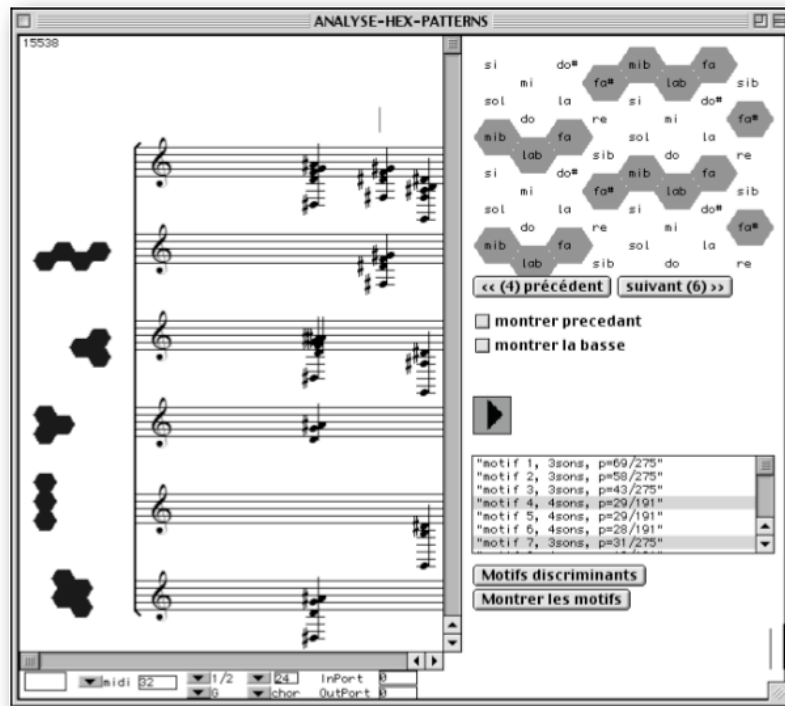
- Une mesure de dissimilarité
- Une matrice de dissimilarité
- Un algorithme de classification ascendante hiérarchique.
- Une représentation visuelle de type « Clustering »

Deux distances ont été testées : la comparaison de texture inventée par Julio Estrada (Estrada 1984) et le vecteur d'intervalles issu de la set-theory d'Allen Forte (Forte 1973).

Les résultats expérimentaux ont permis de poser des repères dans un corpus foisonnant, mais ont montré des limitations liées au caractère statique de la méthode : les enchaînements et donc le temps n'étaient pas pris en compte.

L'expérience d'imagerie visait à dépasser ces limitations. Une représentation dans laquelle les hauteurs sont positionnées dans un réseau hexagonal dessiné sur un tore a été choisie. Proposée par J.M. Chouvel, cette représentation (Chouvel 2005) est inspirée de travaux antérieurs et notamment de l'espace bi-dimensionnel tierces/quintes de Longuet-Higgins (Longuet-Higgins 1979). Sur ce tore, les axes induits représentent les cycles de tierces majeures, mineures, de quintes. La propriété particulière de ce réseau est de faire correspondre un voisinage topologique à un voisinage harmonique. Un accord typique est représenté par un motif géométrique typique.

L'ergonomie de ce prototype met en relation immédiate la partition jouée/visualisée avec une animation dans le tore, ce qui permet d'utiliser les récurrences géométriques et les relations topologiques (inclusions de sous-motifs) comme indices sur les structures d'accord et leurs enchaînements. Un concept a émergé de l'expérience, celui de sous-motif discriminant, ensemble minimal de motifs discriminant les accords d'une séquence. On en déduit une mesure de complexité des accords, et une première compréhension de la dynamique de construction des enchaînements. Ainsi dans une séquence très « atonale » les sous-motifs discriminants peuvent-ils être tonals, ce qui indique une dynamique tonale « souterraine » masquée par une ornementation harmonique : c'est un premier pas vers le décryptage du «mystère Hermeto Pascual».



L'harmonie de Hermeto Pascual disséquée dans OMiel à l'aide d'une vue topologique des familles d'accord

10.4. Extraction de la pulsation, de la métrique et des motifs polyphoniques

Benoît Meudic, thèse de doctorat de l'université Paris 6, 2004.

Ce travail de thèse de Benoît Meudic est basé sur la recherche de structures dans la musique. Il intervient pour une partie dans le cadre du projet européen Cuidado.

Dans un premier temps, la recherche s'est focalisée sur l'extraction de structures rythmiques à partir de fichiers MIDI non quantifiés sujets à des variations de tempo. Le rythme est en effet un composant essentiel de la musique. Sa connaissance est utilisée pour caractériser et décrire une musique, ou un style musical, ou même pour analyser des structures musicales de plus haut niveau (motifs, phrases, mélodies, rythme harmonique). Le principal problème d'une analyse automatique des structures rythmiques vient du caractère perceptif et immatériel de ce phénomène. Le rythme, avant d'être écrit, est quelque chose de perçu. Pour établir un modèle, il faut donc rechercher le rythme non seulement dans la musique mais aussi dans les processus très complexes mis en œuvre par notre perception.

Dans un deuxième temps, la recherche s'est portée sur l'analyse de structures musicales de plus haut niveau que l'on pourrait regrouper sous l'appellation « patterns ». Ces structures sont déterminées par des figures rythmiques, mélodiques ou harmoniques analysées dans leur évolution temporelle. L'idée sous-jacente à cette recher-

che est que ces figures sont répétées plusieurs fois (avec ou sans variations) dans la séquence musicale ce qui donne un critère de base pour leur extraction.

Concernant l'extraction de structures rythmiques, le travail a porté sur le développement de deux algorithmes et leur intégration dans la librairie « Kant » d'OpenMusic. Le premier algorithme recherche les pulsations dans la musique. A partir d'une analyse des premières secondes du morceau, l'algorithme induit plusieurs pulsations possibles, puis cherche à les propager dans le reste de la séquence. Afin de le guider dans sa recherche, chaque événement musical (notes ou accords) est pondéré selon son importance rythmique. Les critères de pondération dépendent du registre, de l'ambitus, de la durée ou de la dynamique de l'événement musical en question. Au final, les différentes pulsations qui ont été propagée sont classées suivant différents critères de vraisemblance. Le deuxième algorithme cherche à grouper les pulsations obtenues en ce qui pourrait correspondre à des mesures. Pour cela, chaque occurrence temporelle d'une pulsation est pondérée, puis une mesure par auto-corrélation permet d'obtenir les groupements les plus probables. Ces deux algorithmes ont été intégrés à la librairie « Kant » d'OpenMusic.

Concernant la recherche de patterns, le travail a consisté à établir un algorithme de mesure de similarité entre deux séquences musicales. En effet, la considération de la simple répétition exacte ne peut suffire, puisque très souvent des modifications interviennent entre l'apparition par exemple d'un thème, d'une phrase ou d'un motif et leurs différentes répétitions. L'algorithme développé mesure la similarité suivant plusieurs critères (contour, intervalles mélodiques, rythme) puis calcule une valeur globale regroupant les différents points de vue. L'aspect le plus intéressant de l'algorithme est que des séquences polyphoniques peuvent être comparées. Par ailleurs, nous n'introduisons pas de connaissances a priori sur le type de musique analysée (musique tonale, atonale...). Ceci donne la possibilité d'analyser un corpus musical assez varié.

La représentation principale des résultats de l'analyse des répétitions se fait par une matrice de similarité qui contient les différentes valeurs de similarité entre toutes les paires possibles de sous-séquences contenues dans le morceau étudié. Un système de couleur permet de visualiser globalement les similarités sur toute la pièce, soit pour chaque point de vue (rythme, contour, ou intervalles mélodiques) soit pour une combinaison donnée des points de vue.

A partir de cette matrice, il est possible de visualiser différents types de répétitions par l'observation des différentes diagonales de même couleur. Certains motifs peuvent alors être extraits, soit parce qu'ils répondent à des caractéristiques de taille (motifs les plus longs ou les plus courts), soit par leurs caractéristiques de fréquence d'occurrence (motifs les plus souvent répétés) [Gomez, Klapuri, Meudic 2003] [Gouyon, Meudic, 2003].

Il est aussi possible de comparer des séquences appartenant à des morceaux différents, et ainsi d'établir des liens entre différentes musiques. Cette opération, contrairement à la précédente qui nécessite par le nombre élevé de comparaison un temps de calcul assez long, ne requiert pas l'extraction préalable de la pulsation, et permet la détection de similarités entre séquences de tailles différentes. Des découvertes intéressantes sur le thématisme trans-œuvres chez Bartok ont ainsi pu être présentées [Meudic, St James 2003] [Meudic 2002].

Plusieurs applications peuvent être envisagées. L'outil peut intéresser : les mélomanes en quête d'une compréhension différente de l'œuvre, les musicologues pour une investigation détaillée d'un grand corpus d'œuvres, les musiciens pour une classification automatique des différentes musiques qu'ils utilisent suivant leur structure musicale (en particulier les musiciens de jazz).

10.5. Analyse computationnelle de partitions selon un modèle d'écoute

Olivier Lartillot-Nakamura. THESE de DOCTORAT de l'UPMC, 2004.

Dans cette thèse a été développée une méthode informatique de recherche automatique de répétitions au sein de partitions musicales. La musique est en effet régie suivant le principe de répétition : l'expression musicale procède par une reproduction de schémas, tandis que l'écoute consiste en une reconnaissance de ces schémas. C'est un des objectifs principaux de la musicologie que de déterminer les schémas caractéristiques de styles musicaux sous forme de théories musicales, ainsi que de mettre en évidence leur présence au sein de chaque œuvre, par l'intermédiaire de l'analyse musicale.

Une systématisation de l'analyse musicale offre deux avantages. D'une part, là où une entreprise humaine laisse une large part aux intuitions implicites de l'analyste, l'explicitation des procédures mises en œuvre — aussi restreintes soient-elles — assure une objectivité des résultats obtenus. D'autre part, une mise en œuvre informatique rend possible une automatisation de la démarche, et ouvre l'accès à un niveau supérieur de détail, tendant vers une certaine exhaustivité.

En opposition à l'approche structuraliste, qui suppose l'existence d'une structure immanente et considère la procédure de découverte de cette structure comme secondaire, il est soutenu que l'œuvre musicale ne peut être saisie, de manière générale, que suivant la démarche de l'écoute. C'est alors la dimension invariante de l'écoute, responsable de l'illusion de la structure immanente, qui rend possible cette systématisation.

L'approche proposée envisage l'écoute sous la forme d'une mémorisation progressive et implique donc un modèle cognitif sous-jacent. L'analyse consiste en une réactivation, suivant deux modes d'accès, des éléments de la mémoire : l'accès syn-

tagmatique rend compte de l'articulation du discours musical sous la forme d'une succession de relations entre notes successives, alors que l'accès associatif réactive les éléments partageant une même identité définie suivant certains paramètres. Le motif musical, c'est-à-dire la répétition séquentielle, résulte alors de l'articulation de ces deux modes d'accès : chaque extension successive d'un motif résulte d'une mise en rapport associatif des extensions syntagmatiques. Une telle modélisation du motif musical est optimisée en temps, car le rapport associatif, qui prend la forme d'une table de hachage, dispense de tout parcours de la mémoire. De plus, grâce à une telle modélisation du motif musical, chaque extension successive d'un motif peut être définie suivant une dimension musicale propre.

Sur la séquence de notes se construit ainsi un réseau de motifs, subsumés à des classes motiviques, lesquelles forment une structure arborescente. Au sein de ces deux réseaux peuvent être tissées des relations de spécificité, rendant compte en particulier des inclusions entre motifs. Ces graphes de spécificité permettent de plus une minimisation des deux réseaux, par annihilation des structures directement déductibles d'autres structures plus détaillées. Cette minimisation assure une optimisation en espace, tout en rendant compte d'une orientation naturelle de la mémoire vers le plus spécifique. Elle permet en outre de remplacer la notion de segmentation musicale au profit du concept de groupement optimal superposable. Enfin, la combinatoire que peut provoquer la répétition successive de classes motiviques est maîtrisée à l'aide d'un mécanisme de rebouclage des classes [Lartillot 2002a 2002b 2002c].

La nouvelle librairie du logiciel OpenMusic (OMKantHus) qui résulte de cette investigation offre des résultats d'une grande fiabilité, qui semblent concorder, pour une grande partie, avec la logique musicale, telle qu'elle est mise en oeuvre par l'écoute. Les capacités combinatoires offrent à la fois une échelle de détail extrêmement fine, et une investigation de longue haleine sur une longue séquence.

Une telle analyse, contrairement à la démarche intuitive humaine, peut ainsi être menée de manière exhaustive. D'autre part, le faible temps de calcul de ces analyses permet d'envisager une application dans le cadre de bases de données musicales. Cet outil informatique suggère ainsi la possibilité d'une explicitation et d'une systématisation de l'analyse musicale, ainsi que d'une modélisation de certains principes mis en jeu lors de l'écoute musicale.

La thèse d'Olivier Lartillot a permis de mettre en évidence, à travers une investigation dans les domaines musicologiques, linguistiques, cognitifs et épistémologiques, l'idée d'un nécessaire conditionnement de l'analyse automatique par des heuristiques perceptives, sous la forme d'une systématisation des mécanismes d'écoute musicale. [Lartillot, St James 2003] [Lartillot 2003a 2003b]

10.6. Segmentation et reconnaissance de patterns : perception, analyse, modélisation.

Ce projet a été réalisé dans le cadre d'une collaboration entre l'équipe Représentations musicales, le Centre Interdisciplinaire pour la Recherche sur la Musique, les Médias et la Technologie (CIRMMT), la Faculté de Musique de l'Université McGill à Montréal et le Département of Music, University of Jyväskylä, Finlande. Il a bénéficié d'un financement de l'Action Concertée Incitative du CNRS « Systèmes complexes en sciences sociales » et était coordonné par Stephen McAdams. J'en coordonnais la branche informatique musicale.

Le projet fait suite aux thèses d'Olivier Lartillot (qui a après sa thèse été recruté en postdoc à l'université de Jyväskylä) et au programme de recherche de Mondher Ayari, entamé dans l'équipe Perception et Cognition et continué dans mon équipe, sur la perception interculturelle : culture et cognition musicale. Les expérimentations liées à ce projet ont eu lieu sur un double corpus oriental-occidental. L'objectif principal était de rendre compte, dans les processus de perception musicale, de l'interaction étroite entre les critères du sujet musicien (interprète et auditeur) et le phénomène d'acculturation musicale. Un aspect particulier du projet concerne l'exploration théorique des relations entre reconnaissance et segmentation, c'est-à-dire des articulations entre pertinences paradigmatiques et structuration syntagmatique, dans la construction de la représentation des flux temporels musicaux.

Le deuxième volet de recherche était centré sur la modélisation cognitive des mécanismes de perception et de compréhension musicales, afin d'explorer en particulier l'interaction complexe entre les processus de segmentation et ceux de la reconnaissance de patterns dans l'écoute musicale. Une des hypothèses du projet est que l'auditeur, pour analyser la structure du Maqâm improvisé (mode musical arabe), engage des processus de segmentation qui opèrent à la fois sur la base d'informations sensorielles de façon « ascendante » et sur la base de connaissances acquises sur les notions théoriques du système musical de façon « descendante », allant d'un pattern d'événements bien connu (motif mélodique, geste de clôture, etc.) à des formes d'enchaînement de sections plus complexes. Il y aurait ainsi une interaction dynamique entre l'analyse des données d'entrée et l'influence des connaissances acquises (le savoir culturel) sur la perception et l'organisation de la structure musicale [Ayari & McAdams 2005]. Il s'est agit d'aborder cette complexité des points de vue de l'analyse musicale, de la perception et de la modélisation informatique en termes de découverte de patterns musicaux pour comprendre l'émergence de la compréhension musicale en temps réel. Le projet a donc mis en regard une modélisation par système complexe d'œuvres musicales avec des hypothèses analytiques et perceptives, dans un double échange avec l'expérimentation : fournir un cadre d'analyse-synthèse propice à la validation d'hypothèses et extraire des indices pertinents propices à l'en-

richissement du modèle. L'ensemble a été appliqué à la perception et cognition musicales des auditeurs occidentaux et arabes à l'écoute de musiques occidentales (Jazz-modal) et arabes (musiques modales improvisées) [Lartillot, Ayari, Assayag & McAdams 2004].

Nous avons poursuivi l'expérimentation comportementale avec, d'une part, un style de musique tunisienne « moderne » qui pose des problèmes de segmentations et de perception de la structure pour les musiciens acculturés, et d'autre part, du Jazz modal selon un schéma qui puisse permettre de rapprocher les deux musiques au niveau de la complexité structurale. Le même paradigme expérimental est appliqué de façon à croiser la musique avec l'origine culturelle des auditeurs. Le but était de mettre en évidence des différences de traitement modaux en fonction des schémas culturels acquis et d'informer les modifications successives des modèles informatiques. Cet aspect a été exploré avec la collaboration de Séverine GABRY dans le cadre de son stage de Maîtrise à l'Université de Reims dirigé par Mondher Ayari dans mon équipe.

En ce qui concerne la modélisation, nous avons utilisé les outils issus de la thèse d'Olivier Lartillot. Des motifs non-pertinents ont été engendrés par ce modèle, en raison de certaines insuffisances. D'autre part, certains motifs importants dans la musique tunisienne « classique » n'ont pas pu être découverts automatiquement, chaque occurrence étant ornementée de manière propre, par l'ajout de notes supplémentaires. Nous avons donc lancé avec Olivier Lartillot un travail visant à comprendre le mécanisme qui permet de ramener la séquence musicale ornementée en une représentation réduite, sur laquelle peuvent être découverts les motifs manquants. La comparaison de cette analyse automatisée acculturée avec une analyse experte avait pour but de mettre en évidence la nécessaire incorporation, au sein de la modélisation de principes stylistiques non décelables directement à partir de la structure de l'œuvre [Lartillot & Assayag 2006]

Après un effort concernant la modélisation en 2005 ayant donné lieu à plusieurs publications [Ayari 2005] [Ayari & McAdams 2005] [Lartillot & Ayari 2005], en 2006 ont été menées les expériences d'écoute et de segmentation sur les musiques de genre Tba' tunisien avec des sujets appartenant aux deux cultures [Ayari & Lartillot 2006].

Les expériences menées en France et en Tunisie avec 80 auditeurs de cultures musicales différentes ont révélé plusieurs indices perceptifs fondamentaux pouvant provoquer une segmentation lors de l'écoute musicale, indices regroupés en trois classes : 1. des indices de surface qui dépendent de façon minimale de l'acculturation, comme les discontinuités entre les attributs auditifs (silence, changement de registre, etc.), la détection de répétitions et le parallélisme

2. des indices portant sur la stabilité relative des événements musicaux par rapport à une hiérarchie fonctionnelle entre les attributs, qui sont mieux détectés chez les sujets acculturés.

3. les groupements en termes selon les échelles modales employées, les patterns spécifiques à un mode musical et enfin, les schémas formels et stylistiques, qui ne sont correctement identifiés que par les sujets acculturés.

Concernant les sujets occidentaux non acculturés dans le style d'improvisation Tba', deux groupes furent constitués : des sujets « ordinaires » et des sujets pratiquant le Jazz. Au vu des résultats, on remarque une difficulté commune aux deux groupes à pouvoir anticiper le cours du développement mélodique et de ses variations modales. Cependant, les musiciens de Jazz obtiennent malgré tout de meilleures performances : une connaissance des échelles modales — acquises par la pratique du jazz-modal, en particulier — et des mécanismes de l'improvisation, bien que différents sur certains éléments, leur permettent de mieux comprendre le jeu musical de l'interprète, et d'analyser plus précisément les schémas de construction mélodiques de l'improvisation. L'expérience dite de « réduction mélodique » donne cependant de mauvais résultats pour les deux groupes non acculturés, à la différence des sujets familiers du Tba' tunisien qui répondent avec cohérence. Cette cohérence de réduction chez les sujets acculturés suggère que la forme mélodique essentielle (la « mélodie mère ») est une représentation importante sur le plan cognitif et qu'elle joue un rôle fondamental dans la segmentation comme processus top-down issu d'une connaissance acquise sur le système musical. Cette hypothèse est étayée non seulement par la convergence des réductions chez les auditeurs acculturés, mais également par la grande divergence de réductions pratiquées par les auditeurs occidentaux qui ne perçoivent pas la construction de l'improvisation selon ce processus. Ces résultats donnent une indication précieuse sur les limitations de la modélisation proposée par O. Lartillot qui ne prend pas encore en compte ces données interculturelles et ne peut exploiter à ce stade que les classes d'indices 1 et 2 [Lartillot 2006].

10.7. Mathématique et Musique : approche algébrique des théories musicales

J'ai organisé en 1999 à la demande de la Société Européenne de Mathématique le Quatrième Forum Diderot « Mathématique et Musique », publié ensuite chez Springer [Assayag, Rodrigues & Feichtinger 2002]. Cet événement distribué sur trois capitales européennes (Paris, Lisbonne et Vienne) comportait notamment une table ronde en visioconférence (en collaboration avec le CNET) impliquant les trois sites. Le Forum « Mathématique et Musique » a eu une influence importante et durable en ce qu'il a renoué les liens entre tous les acteurs du domaine à l'échelon international et redonné une impulsion décisive à l'organisation de la communauté « Math Musique ». Une des actions concrètes qui ont suivi le Forum a été le séminaire MaMuPhi

— Mathématique, Musique, Philosophie — que j’ai monté avec François Nicolas. Parallèlement j’accueillais dans mon équipe Moreno Andreatta pour son stage de DEA « Musique et musicologie du XX^e siècle » poursuivi par une thèse dans la formation doctorale « Musique, Histoire, Société » organisée par l’Ecole des Hautes Etudes en Sciences Sociales (EHESS), l’Institut de Recherche et Coordination Acoustique/Musique (IRCAM), l’Ecole Normale Supérieure (ENS) et le Conservatoire National Supérieur de Musique et de Danse de Paris (CNSMDP). Moreno Andreatta a joué durant sa thèse et ensuite un rôle de premier plan pour capitaliser dans l’équipe et au niveau international les fruits du Forum Diderot.

La thèse de Moreno Andreatta « Méthodes algébriques dans la musique et la musicologie du XX^e siècle : aspects théoriques, analytiques et compositionnels » a été soutenue le 12 décembre 2003 devant un jury interdisciplinaire composé notamment par Alain Poirier (Directeur du CNSMDP), John Rahn (Université de Washington à Seattle), Guerino Mazzola (Université de Zürich) Jean Petitot (EHESS, Ecole Polytechnique), Marc Chemillier (U. Caen) et moi même. Le domaine des liens entre mathématiques et musique y est traité en développant un triple point de vue sur la musique et la musicologie du XX^e siècle.

Un premier aspect concerne la recherche théorique sur les fondements algébriques de certaines structures musicales, recherche menée en collaboration avec des mathématiciens notamment Emmanuel Amiot (autour de la notion de pavage en musique), Harald Friepertinger (sur l’énumération des canons rythmiques à l’aide de la combinatoire algébrique), Thomas Noll (généralisation du modèle de canon rythmique dans le cas de groupes différents du groupe cyclique), Guerino Mazzola (pour la réécriture de la théorie de Vuza et de la *Set-Theory musicale* dans le cadre de la théorie des catégories), Franck Jedrzejewski (pour le modèle algébrique des gammes microintervalliques). Ces travaux sont rassemblés dans un ouvrage de référence, *Perspectives of Mathematical and Computer-Aided Music Theory*, Publié par l’Université d’Osnabrück [Andreatta 2003].

Deuxièmement la thèse développe une réflexion sur les applications analytiques des théories algébriques, avec une généralisation de la *Set Theory* américaine (aux groupes affines et symétriques) implémentée dans la librairie « OMGroups » en OpenMusic. Ce travail a été le prétexte à l’organisation du Colloque International « Autour de la Set Theory » (Ircam, Résonances, 15-16 Octobre 2003) [Andreatta, Bardez, Rahn 2008].

Le troisième axe de recherche concerne l’application compositionnelle des théories algébriques, à partir de l’étude des compositeurs Anatol Vieru, Iannis Xenakis et Milton Babbitt. La thèse a mis en évidence plusieurs points de contact entre la théorie modale d’Anatol Vieru (en particulier la théorie des suites périodiques et des canons rythmiques de pavage), les techniques algébriques de Iannis Xenakis (en particulier les processus de Fibonacci généralisé à valeurs dans des groupes, utilisés par

le compositeur dans la pièce *Nomos Alpha*) et le sérialisme généralisé de Milton Babbitt (en particulier l'isomorphisme entre domaine des hauteurs et domaine rythmique). Les modèles implémentés ont donné lieu à une utilisation en composition contemporaine par les compositeurs (G. Bloch, F. Lévy).

Pendant et après sa thèse, Moreno Andreatta a prolongé ses activités de recherche par un travail intensif de constitution d'un réseau d'experts internationaux dont l'équipe Représentations Musicale constitue le centre de gravité. Cette action s'est articulée autour de l'organisation du séminaire MaMuX (Mathématique/Musique et relations avec d'autres disciplines, faisant suite au séminaire MaMuPhi et impliquant l'Ecole normale supérieure de la rue d'Ulm avec l'Ircam) et du lancement d'une nouvelle revue internationale, le *Journal of Mathematics and Music*, dont Moreno et moi sommes membres fondateurs, suivi de la fondation de la société savante internationale *Society for Mathematics and Computation in Music* (SMCM).

Le séminaire a été un catalyseur important qui a permis d'identifier une communauté et de susciter des collaborations et des publications [Andreatta & Schaub 2003] [Andreatta 2004] [Agon, Andreatta, Assayag, Schaub 2004][Andreatta 2006] [Amiot, Agon, Andreatta 2006] [Ahn, Agon, Andreatta 2007] [Amiot, Agon, Andreatta 2007]. En particulier, les travaux entamés sur la notion de pavage en musique ont pu être approfondis avec l'aide des mathématiciens Emmanuel Amiot, Dan Tudor Vuza, Thomas Noll et Frank Jedrzejewski, permettant notamment une généralisation algébrique à travers de nouveaux algorithmes de factorisation de groupes [Andreatta, Vuza 2001] [Andreatta, Noll, Agon, Assayag 2001]. Moreno Andreatta et Guerino Mazzola ont proposé une contribution à la généralisation de la théorie transformationnelle dans le cadre de la théorie des catégories [Mazzola, Andreatta 2006] dans laquelle ils montrent que les réseaux de Klumpenhouwer, outil très sophistiqué de la théorie transformationnelle américaine, sont en réalité des exemples élémentaires de « limite », au sens de la théorie des catégories.

La réussite au concours d'entrée au CNRS et à l'affectation de Moreno Andreatta dans l'UMR 9912 (STMS) en tant que chargé de recherche de première classe en Octobre 2004 m'a permis de renforcer et d'institutionnaliser l'axe de musicologie mathématique et computationnelle dans l'Equipe Représentations Musicales, qui est devenue à partir de là ainsi une référence internationale dans le domaine des relations Mathématiques et Musique.

Une première thèse entamée dans ce cadre a été « L'hypothèse mathématique chez Xenakis et Babbitt », thèse de Stephan Schaub à Paris IV en collaboration avec le MINT-OMF, Ecole doctorale Concepts et langages.

La recherche porte sur la formalisation mathématique et sur son inscription dans des démarches prospectives en composition musicale. Elle intègre une composante historique, prenant appui sur des approches développées au tournant de la seconde

guerre mondiale, à une recherche de nature théorique basée, d'une part, sur l'étude des éléments formels engagés dans ces approches, et d'autre part, sur leur confrontation avec les œuvres auxquels ces éléments se rapportent.

Les démarches de deux compositeurs ont été étudiées dans le cadre du travail de thèse. Il s'agit de l'américain Milton Babbitt (1916) et du français d'origine grecque Iannis Xenakis (1921-2001). Les approches de ces deux compositeurs représentent deux paradigmes bien distincts, prenant appuis sur des traditions intellectuelles très différentes. Le premier s'approprie les concepts du positivisme logique (cercle de vienne), alors que le second s'inspire, à partir de la fin des années 1950, du structuralisme et du formalisme mathématique (Piaget, Bourbaki). Si ces éléments conditionnent, pour une part, les approches formelles, les projets musicaux respectifs tiennent une place tout à fait décisive. Babbitt reprend les prémisses dodécaphoniques proposées par Schoenberg et en fait son principal objet d'étude. Xenakis, à l'opposé, se détourne du dodécaphonisme et vise une refonte radicale des moyens compositionnels. [Agon, Andreatta, Assayag, Schaub 2004] [Schaub 2005].

La première étape de la recherche a consisté à dégager les éléments formels directement impliqués dans les travaux compositionnels respectifs des deux compositeurs. De les isoler, en quelques sortes, des contextes intellectuels spécifiques auxquels il se rapportent. Cette étape a été menée sur la base des textes théoriques publiés, ainsi que sur l'étude de sources non publiées (documents d'archive). Elle a abouti à la définition d'univers théoriques propres, intégrant divers éléments formels, dont une part peut être intégré à une recherches purement formelles ou faire l'objet de modélisation informatique.

La seconde étape a plus spécifiquement pris en compte les modes d'applications, en tentant de dégager les systèmes et procédures compositionnels. L'analyse d'œuvres y a occupé une place centrale. Quatre œuvres ont fait l'objet d'études détaillées :

- De Xenakis : *Herma* pour piano (1961), *Akrata* pour orchestre à vents (1964-1965).
- De Babbitt : *Three Composition for Piano* (1947-1948), *Semi-Simple Variations* pour piano (1956).

Diverses stratégies ont été appliqué en ces études. Certaines ont directement impliqué l'outil informatique. La modélisation, notamment, de procédures compositionnelles élaborées par Xenakis a fourni un apport heuristique et permis la validation du modèles sous-tendant la composition d'Akrata, reconstitué à partir de documents d'archive. Un système de recherche automatique d'agréats dodécaphonique a également fournie une base heuristique à l'analyse des œuvres de Babbitt.

11. Du rythme

11.1. KANT : Une critique de la quantification pure.

Véronique Verière, Stage de DESS d'Intelligence Artificielle PARIS VI, 1993.

Carlos Augusto Agon Amado, DEA d'informatique de l'université Paris XI. 1994.

Benoît Meudic. DEA ATIAM, Ircam - Université d'Aix-Marseille II, 2000.

A la suite du prototype réalisé en 1993, intégrant notamment le travail de DESS de Véronique Verière, le travail sur Kant a été finalisé par Carlos Agon dans le cadre de son stage de DEA d'informatique de l'université Paris XI Orsay [Verière 1993] [Agon 1994].

Ce dernier a développé un éditeur graphique interactif sous PatchWork [Agon, Assayag, Fineberg, Rueda 1994] dans lequel l'utilisateur contrôle le processus de quantification rythmique en trois étapes : découpage de la séquence en « archi-mesures » (groupes de mesures partageant le même tempo), découpage d'une archi-mesure en segments selon une combinaison de méthodes automatiques et d'interaction manuelle, quantification finale sous la forme d'un affichage en notation musicale (Assayag, Bloch 1997). Le travail théorique a porté sur la définition d'un algorithme de recherche de patterns « flous » : cette technique permet de spécifier des gabarits de figures rythmiques résultats de la combinaison libre d'accélération, de décélération, de notes égales, avec un facteur d'approximation. Le programme propose ensuite une segmentation déduite du positionnement des patterns dans la séquence. Outre les patterns, les méthodes de segmentation disponibles sont les suivantes : recherche de maxima et de minima dans le profil des durées, recherche de groupements déduits de variations brutales de la densité, détection de quasi synchronisme des temps d'attaque dans le cas des séquences polyphoniques, détection des silences. Toutes ces segmentations sont éditables et combinables entre elles par des opérateurs booléens (union, intersection). Dans l'étape de quantification finale, les paramètres de contrôle (contraintes sur les subdivisions du temps, précision etc.) sont maintenant définissables graphiquement au niveau de chaque mesure et de chaque temps. Enfin, Kant est extensible : par la définition en Common Lisp d'une simple méthode, l'utilisateur peut ajouter ses méthodes personnelles de segmentation.

Implémenté par la suite en lisp sous forme de librairie OpenMusic par Benoît Meudic [Meudic 2000], des fonctionnalités ont été rajoutées notamment des méthodes de segmentation :

- repérage de régularités par coefficient d'auto-corrélation. Cette méthode de segmentation a été intégrée à l'éditeur Kant afin de repérer les éléments situés sur une occurrence de la pulsation.

- pondération des événements midi de la séquence musicale en fonction de leurs propriétés de durée, de hauteur, de dynamique..., afin de leur attribuer une hiérarchie qui sera mise en relation avec la forme rythmique. C'est la coïncidence des différentes propriétés qui déterminera un événement hiérarchiquement fort : un accord parfait, de longue durée et joué *forte* sera ainsi différencié d'une note pointée de faible dynamique. Cette hiérarchie sera ensuite mise en relation avec la forme rythmique, en plaçant par exemple les éléments prégnants en début de mesure.

Enfin, des fenêtres d'interface de l'éditeur Kant permettent de contrôler la phase de quantification à chacun des niveaux hiérarchiques de la séquence (au niveau de la note, du groupe de notes défini par la segmentation ou au niveau de toute la séquence). Une gestion complète de la polyphonie est proposée.

12. Opérateurs sur les structures rythmiques dans OpenMusic

Olivier Delerue, DEA ATIAM, Ircam - Université Paris VI, 1997.

Une des faiblesses de PatchWork était son manque de représentations cohérentes et d'opérations sur le rythme. Une des raisons de cette insuffisance provient du fait que le travail sur le domaine du rythme soulève des problèmes algorithmiques beaucoup plus importants que pour le domaine des hauteurs. C'est un des points sur lesquels OpenMusic voulait innover. Aussi, pour répondre à une véritable demande des utilisateurs de logiciels d'aide à la composition, nous avons essayé de définir un ensemble cohérent d'opérateurs rythmiques qui ont fortement mis à l'épreuve les structures de données initialement choisies dans la conception d'OpenMusic : à ce sujet, une attention particulière a été apportée à l'opération de fusion entre des objets musicaux [Delerue 1997] [Agon, Assayag, Delerue, Rueda 1998] [Agon, Assayag, Haddad 2002].

Une fois ces bases établies, nous avons proposé un ensemble de primitives qui mettent en valeur certaines des améliorations attendues, expérimentant ainsi quelques manipulations rythmiques dans le domaine purement symbolique : qu'il s'agisse de fonctions facilitant la transition des objets structurés dans OpenMusic vers leur représentation en notation musicale, ou de transformations musicales directement destinées aux utilisateurs, ces opérations se distinguent par leur délicatesse à procéder sans démolir la structure des objets manipulés.

Des exemples musicaux obtenus à l'aide de ces primitives ont soulevé des problèmes difficilement résolubles dans un cadre direct et complètement déterministe : imposé par la nature même de l'écriture musicale, l'aspect fortement subjectif et contextuel de ces problèmes suggère intuitivement des résolutions basées sur des ensembles d'heuristiques.

Un des résultats les plus intéressants de ce travail a été l'opérateur de *fusion* de structures musicales, problème particulièrement délicat dans son instanciation rythmique notée, qui a donné à OpenMusic une de ses caractéristiques les plus séduisantes.

On notera aussi :

- Régularisation : interprétation de données temporelles et représentation en notation musicale. L'objectif de cet opérateur est d'interpréter des données obtenues de manière arbitraire (algorithmique par exemple) sous forme d'une série de durées et positions temporelles et d'en fournir une représentation en notation musicale. Bien souvent, ces données sont *a priori* trop complexes pour être représentées directement. Plutôt qu'une quantification pure, qui serait soit mal adaptée à la série de données, soit trop contraignante (dégradante vis-à-vis des valeurs originales) l'algorithme recherche un compromis entre un degré de quantification et une certaine complexité d'écriture (imbrications de valeurs irrégulières). La solution proposée est paramétrée par une liste de subdivisions irrégulières autorisées par l'utilisateur.
- Syncopes : cet opérateur, dans la lignée du précédent (dont il réutilise certaines méthodes de découpage hiérarchique à l'intérieur de structures musicales), a également pour but de rendre plus cohérent l'aspect rythmique de certaines structures musicales en forçant, au niveau de la mesure, un découpage temporel suivant les temps déterminés par la signature. Cet opérateur permet donc de supprimer d'éventuelles syncopations qui ne seraient pas forcément souhaitées par le compositeur et complexifient la lecture.
- Quantification : la méthode de régularisation peut, lorsque l'unité globale servant à représenter une structure musicale est une fraction première de la noire par exemple, faire appel à un algorithme de quantification. Le quantificateur réalisé effectue le sur-échantillonnage global d'une structure suivant une unité paramétrable exprimée sous la forme d'une fraction de la noire. Généralement cette unité est choisie plus grande que l'unité actuelle de la séquence, de manière à assurer qu'aucun élément de la structure ne sera détruit.

12.1. Extraction de la pulsation et de la métrique en temps-réel

Thèse de Gilbert Nouno à l'UPMC, 2008.

Ce projet est réalisé dans le cadre de la Thèse de Gilbert Nouno [Nouno 2008] en collaboration avec Philippe Codognet. Il est en relation naturelle avec le projet « Modélisation du style et de l'improvisation » dont il constituera à terme un module. L'objectif est d'étudier et de mettre au point des algorithmes robustes de suivi de tempo/pulsation afin de construire un environnement improvisationnel interactif, mettant en scène des musiciens et un système informatique temps réel. On s'intéresse parti-

culièrement à des conditions rythmiques avancées, où la pulsation perçue n'est pas forcément alignée sur les pics d'énergie. De ce point de vue l'improvisation Jazz est un exemple paradigmatique, qui peut servir de laboratoire pour la mise au point d'algorithmes « tous terrains » applicables dans divers contextes et styles musicaux.

La recherche est basée sur l'hypothèse de mémoire à court terme qui permet au musicien de saisir le tempo après une brève écoute musicale. Les premières étapes du travail ont permis l'expérimentation et la validation de l'algorithme d'estimation de tempo de Simon Dixon (Dixon 2001) à partir de la captation des inter-onset (intervalles entre débuts d'événements). On appréhende l'induction rythmique des impulsions captées à l'aide de classes d'onset définies dynamiquement à partir de fenêtres temporelles variables. Ces fenêtres collectent les impulsions provenant d'une source rythmique (percussionniste ou autre musicien). On approche numériquement la résolution d'une équation aux minimums (comportant une fonction modulo sur les réels) qui permet de donner une grille de « tactus » compatibles. Néanmoins, afin d'aller au-delà de l'approximation numérique, Gilbert Nouno a développé un algorithme permettant de rechercher une grille de coïncidence optimale pour une séquence d'onset. On appelle grille de coïncidence optimale une grille dont la taille du pas et la phase permet de coller au mieux à la séquence pour la quantifier. La thèse montre mathématiquement que ce problème consiste en une optimisation sur un domaine topologique continu bidimensionnel. On peut, grâce à la démonstration d'un théorème, réduire ce domaine continu à un espace discret dont les points correspondent aux solutions recherchées. En découle un algorithme de complexité $O(n^3p^2)$, n étant le nombre de points de la séquence et p le nombre de points de la grille optimale désirée. Un objet temps réel dans l'environnement Max/Msp implémentant cet algorithme a été réalisé. Partant de la modélisation du tempo par une pulsation uniformément accélérée, on déduit une relation mathématique entre temps symbolique de la partition et temps réel de l'exécution. Cette modélisation réintroduite dans l'algorithme amène une dimension d'ordre supérieur dans la complexité, que l'on essaye alors de résoudre par une approximation. Les solutions apportées par l'algorithme fournissent un ensemble de tempi possibles, dont on veut effectuer le suivi par une mise en œuvre d'inférences bayésiennes. On souhaite réaliser ainsi un suivi, ou mieux une prédiction de tempo : une anticipation utilisable à des fins musicales. Cet algorithme propose une quantification à la volée, qui permet de reconstruire en temps réel une interprétation possible du squelette rythmique sous-jacent. Parallèlement, pour consolider l'expertise musicale nécessaire, on a mené une campagne d'entretiens et de travail avec des musiciens (principalement de Jazz). Ces discussions visent à expliciter leur conception de la perception du tempo (stabilité, swing, modulation etc.). Parmi les participants : Riccardo del Fra, Malik Mezzadri, Steve Coleman, Harry Hoennig, Stephane Galland, Joëlle Léandre, des jeunes percussionnistes et batteurs de Jazz.

Gilbert Nouno a tenté en particulier d'élucider et de traiter dans le modèle la question suivante : quelles sont les modalités et la vitesse d'adaptation à un changement de tempo brusque, connu à l'avance ou non, et comment l'intégrer dans un système interactif ? quelle est l'importance du « background culturel » lié au style musical dans le traitement cognitif de cette difficulté ?

12.2. Le geste dans OpenMusic

Dominique Eav, stage d'ingénieur ICAM Toulouse 2002.

Ce stage d'ingénieur (ICAM Toulouse), réalisé par Dominique Eav en 2002, avec l'expertise musicale de Jean-Luc Hervé en collaboration avec Carlos Agon est un premier essai pour l'introduction des « gestes » en OpenMusic [Eav 2002]. Le geste, tel qu'il a été défini dans ce stage, est un élément qualitatif, décrivant par exemple un « trait » musical (comme une montée chromatique), il représente alors l'évolution de la phrase. Le geste peut également décrire l'évolution de paramètres internes à une note ou à un ensemble de notes, comme un crescendo ou une variation continue de hauteur. L'intégration du « geste » permet d'agrandir le cadre d'application d'Open Music, en permettant par exemple de piloter directement un moteur de synthèse comme CSound ou MAX, les gestes étant utilisés comme des contrôleurs.

Le résultat matériel de ce stage est un code Lisp implémentant principalement : la définition et la visualisation des gestes, leur intégration dans une partition et leur utilisation pour le contrôle de la synthèse. L'intégration de ce code dans OpenMusic reste cependant problématique.

13. Intégration signal-symbolique

13.1. Composer la synthèse

Peter Hanappe, DEA ATIAM, Ircam - Université Paris VI, 1995.

Peter Hanappe, Design and implementation of an integrated environment for music composition and synthesis. Thèse de Doctorat de l'UPMC, 1999.

Lors du stage de DEA de Peter Hanappe [Hanappe 1995], une amélioration décisive a été apportée au logiciel AudioSculpt de l'Ircam (Eckel 1992). Nous avons proposé d'entourer les techniques habituelles de suivi de partiel d'une batterie de procédures de discrétisation (quantification) rythmique et harmoniques basées sur une segmentation automatique ou manuelle du signal. Le contenu spectral borné par ces segments, simplifié et moyenné est alors interprété comme une séquence d'accords, et exporté vers un environnement de composition dans lequel ces derniers peuvent être manipulés comme données symboliques. Cette simple idée a été le premier pas,

à l'Ircam, de ce qui est ensuite devenu le « paradigme » des traitements *signal-symbolique*, un ensemble de passerelles entre le monde du signal et celui des organisations symboliques (hauteurs, durées, rythme, harmonie etc.) [Assayag, Hanappe 1996]. Cet outil a eu un grand succès et a été utilisé dans beaucoup de compositions (par exemple *Locus* de Claudy Malherbe, 1997, ou *Empreintes* de Joshua Fineberg, 1995).

Dans le cadre du projet de thèse qui a suivi, nous avons réfléchi à la possibilité d'un environnement intégré pour la composition et la synthèse musicale sur ordinateur [Hanappe, Assayag 1997] [Hanappe, Assayag 1998]. Cela constituait la suite logique d'une démarche qui commençait à se formaliser sous les auspices des *relations signal-symbolique*.

L'architecture élaborée combine dans un seul système la synthèse en temps-réel, une gestion de la mémoire dynamique avec un ramasse-miettes, le traitement interactif d'événements, et le dialogue avec l'utilisateur à travers un interprète Scheme. L'environnement est entièrement écrit dans le langage de programmation Java et peut être utilisé dans des applications réparties. Trois aspects du domaine de la musique sur ordinateur — composition, synthèse et interactivité — sont intégrés de manière intime dans cet environnement. Les concepts et les structures décrits dans cette thèse peuvent être étendus et appliqués plus généralement aux environnements multimédia.

L'interprète Scheme embarqué offre un environnement interactif de programmation. La plate-forme Java sous-jacente promeut l'utilisation transparente des objets fonctionnels dans le système. Ces objets fonctionnels, appelés programmes, sont utilisés pour décrire le comportement complexe du système. Les événements, par exemple, sont composés d'un temps d'évaluation, d'un programme et des arguments du programme. Au temps indiqué le programme est appliqué avec ses arguments. Ainsi, l'interprète permet à l'utilisateur d'exprimer et modifier les actions du système dans un langage de haut-niveau.

L'unité de base de la composition est nommée activité. Elle est composée d'une position dans le temps, d'une durée, d'un programme de début et d'un programme de fin. Les deux programmes décrivent respectivement les actions à prendre pour commencer et terminer cette activité. Cette organisation n'est pas sans faire penser à ce qui deviendra par la suite le concept de *Maquette* dans OpenMusic. L'organisation d'activités dans le temps est prise en charge par une structure appelée pattern. Les patterns regroupent un ensemble d'activités ainsi que les relations temporelles entre elles. Ces relations sont vérifiées après chaque manipulation de temps et une réorganisation des positions et des durées des activités peut en résulter. Cette stratégie se rapproche des techniques de la propagation de contraintes que l'on trouve dans la construction des interfaces graphiques. Les motifs permettent d'insérer un pattern à plusieurs reprises dans une composition. L'organisation des activités de durée in-

connue au moment de la composition peut être réalisée à l'aide des opérateurs d'intervalles basés sur des relations causales.

Les patterns et les motifs permettent la structuration d'une pièce de manière hiérarchique. A chaque niveau de cette hiérarchie est attaché un axe de temps local. L'axe de temps d'un niveau est projeté sur l'axe du niveau supérieur moyennant un modèle de temps. Ces modèles permettent la déformation du temps local.

L'objet unité pour la génération de son est appelé un processus de synthèse. Il est créé en utilisant une stratégie de méta-classes : une technique de synthèse crée une nouvelle voix de synthèse, la voix de synthèse crée un nouveau processus de synthèse. L'information temporelle codée dans la partition est rendue disponible aux processus de synthèse. Cette information est regroupée dans un objet appelé contexte de temps. Il assure, entre autre, la conversion du temps indiqué par le synthétiseur dans le temps local d'un processus de synthèse.

L'interactivité du système est rendue possible grâce à la gestion concurrente de la synthèse en temps-réel, du traitement des événements et de l'entrée de données textuelles dans l'interprète Scheme. Dans cette thèse, Peter Hanappe évalue si un environnement dynamique, intégrant un ramasse-miettes, peut garantir une exécution en temps-réel « dur ». Il est difficile de répondre à cette question seulement à partir des discussions qu'on trouve dans la littérature. En revanche, Hanappe introduit une contrainte imposée sur les processus de synthèse qui réduit l'interaction entre le ramasse-miettes et la tâche temps-réel au problème du scheduling de tâches concurrentes [Hanappe 1999].

13.2. Agencer la synthèse comme un espace

Thèse de doctorat d'Olivier Delerue à l'UPMC. Spatialisation du son et programmation par contraintes : le système MusicSpace. 2004.

Après son stage de DEA dans l'équipe Représentation Musicales, Olivier Delerue a entamé une thèse de doctorat chez Sony-CSL avec François Pachet, en co-direction avec moi. La partie qui concerne plus particulièrement l'équipe Représentations Musicales est l'environnement OpenSpace [Delerue, Agon 1999] en collaboration avec Carlos Agon.

L'exploration en temps réel d'espaces de paramètres arbitrairement grands suppose un moyen ouvert pour définir ces espaces et une interface adaptée pour les exploiter. C'est à ce problème que nous avons tenté d'apporter une réponse en proposant un pont de communication entre le système de contrôle temps réel MusicSpace, développé par Olivier Delerue chez Sony-CSL et le langage visuel de programmation musicale OpenMusic. Grâce à cette connexion (appelée OpenSpace) MusicSpace

permet de contrôler dynamiquement et temps-réel un ensemble d'objets musicaux définis en OpenMusic.

Le système MusicSpace [Delerue 2004] a été conçu initialement pour le contrôle en temps réel de la spatialisation de sources sonores sous contraintes : au sein d'une interface graphique, l'utilisateur a la possibilité de définir la position initiale de chacune des sources sonores à spatialiser, ainsi qu'un ensemble de contraintes sur ces sources sonores qui établissent des règles de comportement entre les différentes sources. L'interface représente graphiquement l'ensemble des sources sonores définies ainsi que, sous forme d'icône, un objet représentant l'utilisateur. Celui-ci a alors la possibilité de déplacer chaque objet (dans un plan azimutal) et de percevoir en temps réel les modifications correspondantes de la spatialisation des sources. Les contraintes permettent alors d'établir des relations entre les sources sonores de manière à préserver une certaine qualité du mixage.

Cette interface présente des possibilités importantes puisqu'elle n'est pas liée à la nature des objets qu'elle contrôle : en particulier, elle n'est pas réduite au contrôle de la spatialisation qui n'en est qu'une application parmi d'autres. De même, l'ensemble des contraintes conçues pour l'application à la spatialisation du son appartient au type de représentation graphique et non à l'application elle-même, ce qui, dans un contexte différent, leur permet de conserver une interprétation valable.

Openspace permet une autre utilisation possible de MusicSpace : le contrôle de la synthèse. A la différence de la spatialisation, celle-ci requiert un nombre plus important de paramètres (et donc d'objets graphiques) dont la génération manuelle est relativement fastidieuse. De plus, une caractéristique essentielle de ces espaces de paramètres est que leur définition (autant la définition même de ces paramètres que celle des relations qui les unissent) possède une forte dominante algorithmique : la conception d'un cas de figure particulier s'exprime le plus souvent, par exemple, par l'intermédiaire de boucles. C'est donc très naturellement que s'adapte un environnement de programmation visuelle pour répondre à ce type de besoin.

OpenMusic va répondre directement à ces deux problèmes en proposant la modélisation d'un tel espace sous la forme d'un patch (programme visuel) : un nombre arbitrairement grand de paramètres peut alors être défini et organisé de manière algorithmique. La communication repose sur l'utilisation systématique du système d'exploitation MidiShare du GRAME qui est utilisé pour l'implémentation des deux systèmes.

Le problème général que pose le contrôle de la synthèse revient d'une part à agir simultanément sur un ensemble arbitrairement grand de paramètres de contrôle, et d'autre part à faire varier ces paramètres de manière cohérente les uns par rapport aux autres : ceux-ci ne sont généralement pas totalement indépendants les uns des autres et donc la variation de l'un d'entre eux impose éventuellement celle de cer-

tains autres. OpenSpace est un environnement ouvert permettant de spécifier de manière algorithmique d'une part, et d'interpréter en temps-réel d'autre part de tels espaces de contrôle. Le cas de la synthèse additive a été particulièrement expérimenté. Dans OpenSpace, on représentera dans l'espace de contrôle des partiels du spectre (visualisés comme des points) créés dynamiquement à l'aide d'OpenMusic, répartis dans des régions de cet espace. Un observateur ponctuel se promène dans cet espace, révélant (sur le plan sonore) les régions qu'il visite. OpenMusic peut aussi créer algorithmiquement des contraintes sur les objets « partiels » : regroupement en régions, sphère d'influence sonore sur l'observateur, filtrage harmonique, comportement particuliers lorsque des groupes de partiels sont déplacés. Enfin, la création des espaces de paramètres, celle des objets de synthèses répartis dans ces espaces et des contraintes qui les relient peuvent être séquencés dans le temps grâce aux maquettes OpenMusic.

13.3. Représenter les données d'analyse

Travail pour le diplôme d'ingénieur de Jean Bresson, ESSI - Ecole Supérieure en Sciences Informatiques, Université de Nice - Sophia Antipolis, 2003.

Ce travail s'inscrit dans le cadre du projet *OMChroma* [Stroppa, Lemouton, Agon 2002] porté principalement par Carlos Agon et Jean Bresson dans mon équipe, et porte sur l'intégration des fichiers SDIF (Sound Description Interchange Format) [Bresson, Agon 2004] dans l'environnement OpenMusic et sur leur utilisation dans d'autres contextes d'analyse.

Le format SDIF, développé conjointement par l'IRCAM et le CNMAT, est un format générique, ouvert, et multi-plates-formes, pour le stockage et l'échange de données de description sonore. De telles données proviennent principalement de programmes d'analyse ou de traitement du son, développés à l'Ircam ou ailleurs. C'est par exemple le cas de *Diphone*, qui produit au format SDIF les résultats d'analyse par modèles de résonance (*ResAn*), ou additive (*AddAn*) (Rodet 1997). Il pourra également s'agir d'enveloppes spectrales, de *FOF* (formes d'ondes formantiques), ou de tous autres types de données.

Le logiciel OpenMusic permet de manipuler les fichiers *SDIF*. L'idée était de réaliser un éditeur correspondant à cette classe objet, au même titre que les éditeurs existant pour les autres objets musicaux dans OpenMusic (les notes, accords, les sons, etc.) Nous avons abordé un des concepts fondamentaux de *Chroma*, visant à effacer la frontière entre le traitement symbolique des structures musicales en CAO (la manipulation de modèles ou objets musicaux), et le traitement du signal sonore (analyse, synthèse). Ces deux domaines distincts se côtoient en effet de plus en plus dans la création contemporaine. Nous essayons par cette démarche de faire des données de description sonores (contenues, dans le cas qui nous intéresse, dans un fichier SDIF)

des objets musicaux à part entière, exploitables à ce titre dans l'environnement de composition. On entend par données de description sonore tout type de données d'analyse d'un son, ou même de données de paramétrage de synthèse sonore. La CAO peut en effet être complémentaire à la synthèse des sons dans le sens où les sons créés sont le résultat de processus logiques dépendant de paramètres de synthèse.

Le fait de disposer d'un éditeur pour les données SDIF dans l'environnement OpenMusic permet au compositeur d'avoir une vue et un réel contrôle sur les données de descriptions sonores qu'il possède. L'objet musical constitué par ces données pourra subir des traitements, que l'on sera en mesure d'apprécier visuellement, et qui se répercuteront sur la suite de l'exécution. Lui-même pourra être le résultat d'un processus en aval. Il ne s'agit donc plus uniquement d'un son que l'on a analysé, mais d'un véritable modèle faisant partie intégrante de la structure des algorithmes entrant en œuvre dans le processus de composition.

Les problématiques de la représentation des données SDIF sont diverses ; la principale étant la grande flexibilité du format de laquelle résulte la nécessité de s'abstenir de tout a priori quant au contenu (nature des données, interprétation, taille, ...) du fichier que l'on va visualiser. Toutefois, il est important d'offrir à l'utilisateur des possibilités de visualisations et d'éditations adaptées à ses besoins. Cette dualité impose donc de concevoir des systèmes de navigation intuitifs et efficaces se positionnant entre la généralité des données et la spécificité nécessaire à une représentation suffisamment explicite et parlante.

L'étude de la structure du format SDIF a mis en évidence l'utilité d'une représentation graphique 3D des données. Celle-ci permet une représentation des différents paramètres ajoutée à leur évolution temporelle, sous forme d'une « grille » tridimensionnelle. Dans cet espace, l'utilisateur peut se déplacer, se focaliser sur des zones d'intérêt, éditer les données selon diverses procédures. Cette représentation générale en 3D a également été complétée par une navigation de plus haut niveau permettant la sélection d'un certain type de données parmi toutes celles que peut renfermer un même fichier, et par diverses études de représentations alternatives, parfois plus adaptées à des types de données spécifiques. On peut ainsi visualiser les données sous forme de courbes, de mini-sonagramme (représentations plus proches de ce qui se fait dans le domaine de l'analyse sonore), ou à l'aide d'un mini-synthétiseur permettant une appréciation auditive des caractéristiques des données.

D'un point de vue plus technique, cet éditeur SDIF présente quelques aspects intéressants. Il s'agit tout d'abord d'une application indépendante, écrite en C++. Sa réalisation a donc nécessité la mise en œuvre d'une communication entre OpenMusic et une application externe, écrite dans un autre langage, ce qui a ouvert largement à de nouvelles possibilités de développement.

Enfin, toute l'application et son interface ont été réalisées en utilisant *OpenGL* et la librairie *GLUT*, ce qui la rend parfaitement portable sur les autres plates-formes. *SDIF-Edit* peut donc être utilisé indépendamment comme éditeur SDIF sous Linux, ce qui représente un intérêt pour des applications dans de nouveaux contextes (autres que la CAO). En effet il n'existait encore aucun éditeur permettant de manipuler les fichiers SDIF.

13.4. Ecrire le son et la synthèse

Thèse de Jean Bresson à l'UPMC, 2006

Ce projet principalement coordonné par Carlos Agon dans l'équipe porte sur l'écriture du son dans le cadre de la composition de musique électroacoustique ou mixte, c'est-à-dire d'une musique ayant recours aux technologies de synthèse et/ou de traitement numérique du signal sonore. Les techniques de synthèse sonores disponibles actuellement permettent d'envisager des possibilités de création quasi-illimitées dans le domaine sonore, mais la complexité des structures et systèmes mis en jeu limitent l'accessibilité de ces possibilités pour les musiciens et compositeurs.

Dans ce contexte, complémentairement aux environnements de contrôle de la synthèse destinés à proposer des interfaces de plus haut niveau pour la synthèse sonore, la composition assistée par ordinateur peut certainement apporter une vision nouvelle sur ce problème. L'approche traditionnellement soutenue par l'équipe Représentations Musicales, notamment à travers l'environnement de composition OpenMusic, considère les structures symboliques et les langages informatiques en relation aux formalismes musicaux, en vue de la constitution de modèles informatiques correspondant à des projets compositionnels. C'est donc l'écriture, en tant qu'activité de modélisation, qui est mise en avant plutôt que les données musicales à proprement parler.

En étendant cette notion au domaine du signal sonore, une nouvelle approche de la synthèse peut émerger, axée prioritairement sur des considérations d'ordre symboliques et compositionnelles, plutôt que sur les problématiques liées au traitement du signal. La question de l'articulation du signal sonore et du signe musical dans la composition musicale et dans la CAO est ainsi l'un des axes principaux de ce projet, visant à explorer des nouvelles modalités d'écriture musicale mettant en jeu les sons et la synthèse sonore.

Différents outils et bibliothèques ont ainsi été réalisés dans OpenMusic par Jean Bresson avec le compositeur Marco Stroppa (qui a initialement conçu l'environnement *Chroma*) pour permettre un travail sur le son et les données de description sonore (outils pour le traitement du signal audio, interfaces avec des programmes d'analyse, de traitement et de synthèse sonore, outils de programmation pour le traitement des données au format SDIF, etc.) L'utilisation de cet environnement compo-

sitionnel étendu (appelé *om-sounds*) en complémentarité avec les outils de programmation disponibles dans OpenMusic, permet aux compositeurs de créer et contrôler des processus compositionnels complets allant de la spécification de paramètres de haut niveau jusqu'au niveau du traitement du signal, tout en maintenant (notamment à travers la manipulation des abstractions) un contrôle symbolique sur ces différents niveaux de complexité des processus.

Parallèlement, des outils pour l'analyse sonore et la structuration des données de contrôle permettent d'aborder, encore une fois à un niveau symbolique, la génération de paramètres pour les processus de synthèse. Une extension de l'environnement OMChroma par des structures mixtes pour la génération, la représentation et la transformation de ces données de contrôle a été implémentée. [Bresson, Stroppa, Agon, 2005] [Bresson, Agon 2006] [Bresson, Stroppa, Agon, 2007].

Enfin, les repères et interactions temporelles entre objets et processus de synthèse sonores ont mis en évidence l'importance des problématiques temporelles, qui rapprochent ce projet d'une autre question transversale des systèmes de composition qui est celle de la gestion et de l'organisation du temps dans les processus musicaux. [Bresson, Agon, Assayag 2006] [Bresson 2006, 2007] [Chouvel, Bresson, Agon 2007] [Bresson, Agon, Assayag 2008].

Les maquettes OpenMusic ont ainsi été remaniées afin de constituer une interface pour l'écriture et la représentation des processus de synthèse sonore en étroite relation avec les structures temporelles. Cette nouvelle maquette constitue une forme de partition ouverte permettant la mise en place d'un système de représentation symbolique et personnalisé pour l'écriture électroacoustique [Bresson, Agon 2007] [Bresson, Agon 2008].

13.5. Orchestration computationnelle

Thèse de Grégoire Carpentier à l'UPMC, 2006-2008.

Ce projet fait l'objet d'une thèse par Grégoire Carpentier et constitue une des composantes du projet ANR « Sample Orchestrator ».

Sensible à la demande insistante des compositeurs de disposer, au sein d'un environnement de composition assistée par ordinateur, d'un outil puissant pour le contrôle et d'exploration du timbre orchestral, l'IRCAM a encouragé il y a quelques années la création d'un groupe de travail réunissant chercheurs et compositeurs sur ce sujet. Il en est ressorti un « cahier des charges » définissant un ensemble de priorités de recherche à court et moyen terme. Devant l'ampleur des travaux qu'un domaine si vaste et si complexe ouvrait, deux thèses de doctorat sur l'aide à l'orchestration (l'une dans l'équipe Représentations Musicales, l'autre dans l'équipe Analyse-synthèse) ont débuté à l'IRCAM durant l'année 2005, et furent conduites en parallèle.

Dans le cadre du projet ANR j'ai coordonné le travail des deux équipes visant à la production d'un environnement logiciel opérationnel pour les compositeurs en production. Cette séparation des tâches entre les deux équipes était indispensable pour aborder le problème de l'orchestration assistée par ordinateur selon différents axes de complexité. Les travaux de Damien Tardieu dans l'équipe Analyse-synthèse ont essentiellement consisté en l'élaboration d'un cadre formel adapté à la caractérisation du timbre instrumental et orchestral. Il s'agissait d'abord d'identifier un ensemble de dimensions perceptives pertinentes pour la description du timbre, puis de trouver un moyen d'extraire automatiquement ces propriétés à partir de grandes banques d'échantillons, pour enfin mettre au point un modèle capable de prédire les caractéristiques acoustiques de mélanges instrumentaux. Ce travail a débouché sur l'élaboration de modèles d'instruments qui permettent d'estimer la probabilité qu'une combinaison instrumentale possède une caractéristique du timbre désiré par le compositeur (Tardieu, Rodet 2007).

Dans l'équipe Représentations musicales, avec Grégoire Carpentier, nous nous sommes concentrés sur les difficultés liées aux aspects combinatoires du problème. L'orchestration est en effet l'art subtil et délicat des alliages instrumentaux, et il suffit de considérer l'éventail de timbres couvert par chaque instrument pour se faire une idée de l'étendue des possibilités sonores offertes par un orchestre. Une telle explosion combinatoire interdisant le recours à des méthodes complètes de résolution, nous nous sommes tournés du côté des métaheuristiques en optimisation multi-objectifs (Ehr Gott, Gandibleux 2000) dont l'utilisation de plus en plus fréquente ces dernières années en optimisation combinatoire et en programmation par contraintes a permis de traiter des problèmes très complexes.

L'objectif est de réaliser un environnement d'aide à l'orchestration utilisant les ressources disponibles en termes de bases de données d'échantillons instrumentaux et de méta-données constituées de descripteurs de ces sons. Dans un contexte de composition assistée par ordinateur, l'aide à l'orchestration a été identifiée comme un problème d'optimisation multi-objectifs sous contraintes. Une telle approche est nécessaire pour tenir compte à la fois du caractère multidimensionnel de la perception du timbre et du nombre potentiellement élevé de contraintes liées à l'écriture pour orchestre. En outre, l'approche multi-objectifs a l'avantage de retourner non pas une unique solution mais un ensemble de solutions dites efficaces, chacune d'entre elles réalisant un compromis différent entre les objectifs à atteindre. Le compositeur a donc le choix parmi différentes écritures conduisant à des timbres variés et impliquant chacune une pondération différente des critères.

Dans un premier temps, la recherche s'est orientée vers une modélisation inspirée du sac-à-dos binaire multi-objectifs (MOKP) et un principe de résolution hybride, combinant exploration génétique (amélioration itérative d'une population de solutions) et heuristiques de recherche locale (amélioration itérative d'une seule solution)

(Jaszkievicz 2001). Issue de cette recherche, une seconde mouture de l'algorithme d'orchestration a été développée et a permis d'obtenir des résultats convaincants et jusque-là inédits, dépassant largement les performances de la première version utilisant une méthode de recherche complète. Cette approche a toutefois rapidement montré ces limites : bien qu'utilisant des opérateurs génétiques conçus dans le but de limiter l'apparition de solutions inconsistantes (au sens d'un réseau de contraintes) dans la population, ces derniers ne possèdent pas la généralité suffisante pour s'adapter à d'éventuelles contraintes supplémentaires. Ainsi, dans le cas d'un problème relativement contraint, les chances de produire des solutions réalisables sont minimales, et l'évolution est compromise [Carpentier, Tardieu, Assayag & al 2006].

La modélisation du problème de l'orchestration a ensuite été entièrement repensée, délaissant l'approche du MOKP pour s'orienter vers un problème d'optimisation combinatoire plus classique, où l'espace des possibilités est exprimé à travers un ensemble de couples variable/valeur. Cette représentation permet notamment de s'affranchir des limitations liées à l'effectif orchestral, contrainte principale et irréductible du problème de l'orchestration. Ainsi, en l'absence de contraintes supplémentaires, le problème devient non-contraint et les opérateurs génétiques classiques (crossover et mutation) reconnus pour leur efficacité exploratoire peuvent alors être utilisés [Carpentier, Tardieu, Assayag & al 2007a et 2007b] [Carpentier, Tardieu, Harvey, Assayag & al 2008]

Une séparation totale a été établie entre la partie liée à l'optimisation d'une part, évaluant la pertinence d'une solution sur la base de ses descripteurs audio, et à la satisfaction des contraintes d'autres part, évaluant la consistance d'une solution sur la donnée de sa description symbolique. L'optimisation est prise en charge par les mécanismes d'évolution propres aux algorithmes génétiques tandis que la satisfaction des contraintes est assurée par une méthode de réparation des éventuelles solutions inconsistantes dans la population. Un formalisme permet d'exprimer une grande variété de contraintes globales sur les descripteurs symboliques, c'est-à-dire sur les paramètres de l'écriture musicale. Une méthode efficace de réparation tirant parti de la distinction entre contraintes de « design » et contraintes de « conflit », a été développée et évaluée avec succès sur un grand nombre de cas. Une troisième version de l'algorithme d'orchestration tirant parti de ces travaux a permis d'appréhender des problèmes d'orchestration plus complexes, plus variés, et pour des formations instrumentales sensiblement plus larges que précédemment [Carpentier 2008]. La qualité des solutions obtenues a été confirmée par de nombreux compositeurs collaborant au projet en proposant de cas d'étude toujours plus complexes. L'outil d'orchestration a en outre été utilisé par le compositeur Jonathan Harvey pour une nouvelle pièce (*Speakings* a été créé le 19 août 2008 au Royal Albert Hall de Londres, interprété par le BBC Scottish Symphony Orchestra, dirigé par Ilan Volkov). Cette collaboration a notamment permis d'entrevoir de quelle manière les

contraintes peuvent servir à contrôler l'évolution temporelle d'un timbre donné [Carpentier 2008].

Jean Bresson a écrit un prototype de contrôle en OpenMusic qui permet de partir d'objets abstraits, tel un accord ou une séquence d'accords et de générer des spécifications de cibles « abstraites » pour le moteur d'orchestration [Carpentier, Bresson 2008]

14. Contraintes et processus

14.1. Objets , contraintes et processus

Le projet AVISPA (Ambientes Visuales de Programacion Aplicativa, Environnement Visuel de Programmation Applicative) a fait l'objet d'une collaboration entre l'Ircam et plusieurs instituts universitaires, et a reçu un financement de Colciencias, l'institut colombien de soutien à la recherche scientifique. Cette collaboration a été grandement facilitée par la possibilité d'accueillir à plusieurs reprises Camilo Rueda, qui a joué un rôle fondamental de coordinateur de ce projet, en chercheur invité à l'Ircam.

Les institutions concernées étaient outre mon équipe :

Université Javeriana (Cali, Colombie) : Camilo Rueda, Francisco Valencia, Gabriel Tamura, Luis Quesada

Université Del Valle (Cali, Colombie) : Juan-Francisco Diaz , Gloria Alvarez

Université De Los Andes (Bogotá, Colombie) : Olga Marino, Silvia Takahashi

Ce projet a une forte composante théorique et prospective et cherchait, tout en confortant notre savoir faire très spécifique en matière de langage informatique pour musiciens, à nous maintenir en bonne position pour relever les défis à venir en cette matière. A partir des travaux pionniers de Camilo Rueda et son équipe à l'université Javeriana de Cali, le consortium cherchait à spécifier formellement un langage visuel d'aide à la composition musicale qui intègre les concepts fondateurs de plusieurs grandes familles de langages de programmation : les Objets (des unités fonctionnelles indépendantes comprenant leurs propres données et leurs propres procédures de manipulation), les Contraintes (des formules logiques définissant les propriétés et les relations qu'entretiennent les objets dans un environnement informatique), et les Processus (des unités de programmes qui peuvent s'exécuter en parallèle et sont donc bien adaptées pour représenter le temps musical, notamment dans un contexte temps-réel). La stratégie générale du projet était de baser le langage sur des modèles formels rigoureusement définis au plan mathématique, aussi bien pour sa sémantique (la description de son fonctionnement) que pour sa syntaxe, notamment

visuelle. Pour fonder notre langage sur un modèle formel solidement établi nous proposons une extension d'un calcul formel (un ensemble d'axiomes et de règles permettant de construire toutes les propriétés intéressantes du système) mis au point par R. Milner et appelé pi-calcul (Milner 1992). Ce dernier permet de modéliser le fonctionnement de processus concurrents (qui s'exécutent en parallèle dans un ordinateur) ainsi que les communications entre ces processus. Dans notre extension, nous rajoutons la notion d'objet (au sens des langages objet défini plus haut) et de contrainte logique entre objets.

Dans une phase préliminaire, nous avons commencé par explorer deux voies pour la construction d'un modèle sémantique de l'intégration objet et contraintes :

- En partant d'une notion standard d'objet et en y ajoutant la possibilité de définir des contraintes à l'intérieur d'un même objet, sur ses attributs. Le calcul d'objets de Abadi-Cardelli (Abadi, Cardelli 1994) a été étendu avec des contraintes sur les attributs. Tout changement de valeur d'un attribut lié par des contraintes déclenche une exploration. Si la valeur satisfait les contraintes alors la réduction normale du calcul d'objets est faite. Dans le cas contraire le calcul échoue.
- En explorant des extensions du calcul CC. La simulation d'objets et de l'héritage dans Oz a été prise comme base (Henz, Smolka & Wurtz 1995).

Dans une deuxième phase, la notion de processus concurrent a été finalement adoptée comme unificatrice d'objets et de contraintes. Cette notion n'existant pas dans le calcul Abadi-Cardelli, le π -calcul de Milner a été choisi à sa place. Dans Oz la notion d'objet n'étant pas primitive, cette piste a été aussi abandonnée.

Le π -calcul a été étendu en paramétrant toute communication entre processus par un « store », zone de stockage des contraintes. Un objet peut désormais être localisé (dans le sens du π -calcul) dans une variable. Les opérations standard ask et tell du modèle CC ajoutent des contraintes sur ces variables et donnent la possibilité de décrire des schémas de synchronisation complexes entre processus. On a appelé $\pi+$ -calcul cette extension.

Le $\pi+$ -calcul a ensuite été étendu avec une notion d'objet similaire à celle de TYCO (Vasconcelos 1994) Dans ce calcul, appelé PiCO (« Pi+Contraintes+Objets ») la notion de classe a été codifiée ainsi qu'une notion simple d'héritage par extension, sans redéfinition de méthodes. Il faut toutefois préciser une limitation : les contraintes doivent s'exprimer sur des variables de la même classe, ou de classes dérivées l'une de l'autre [Alvarez, Diaz, Quesada, Valencia, Rueda, Assayag 1998].

Le langage à composants visuel « Cordial », dont la syntaxe permet de manipuler aisément les entités du $\pi+$ -calcul, a été par la suite prototypé. L'éditeur graphique de Cordial a été programmée en Java et testée sur des plate-formes Linux [Rueda, Tamura, Quesada 1997] [Rueda, Assayag, Laurson, Bloch 1998][Rueda, Alvarez, Quesada, Tamura, Valencia, Diaz, Assayag 2001].

14.2. Contraintes et recherche locale

Thèse de Charlotte Truchet, université Paris 6, 2004.

La Programmation par Contraintes (PLC) est un paradigme de programmation, qui connaît depuis quelques années un succès certain. Une « contrainte » est tout simplement une relation logique portant sur un ensemble de variables (inconnues d'un problème, variant dans un domaine précis). Une contrainte donne une information partielle sur les valeurs que les variables peuvent prendre, « contraignant » ainsi cet espace de valeurs. L'ensemble variables, domaines, contraintes forme un CSP, Constraint Satisfaction Problem. L'idée de base de la programmation par contraintes est d'intégrer le concept de contrainte dans un langage de programmation de haut-niveau pour pouvoir programmer avec cette notion d'information partielle. Une fois le CSP écrit dans un langage précis, il est passé à un solveur qui se charge de trouver des valeurs satisfaisant les contraintes.

La PLC trouve naturellement une place en musique, comme le montrent les nombreux travaux sur l'harmonisation automatique (Ballesta 1998) (Pachet, Roy 2001). En musique contemporaine, une phase de consultation de compositeurs liés à l'IRCAM par Charlotte Truchet [Truchet 2003] a permis de constater le nombre et la diversité des CSPs musicaux. Au total, quatorze CSPs ont été étudiés et résolus, en composition et en analyse musicale, grâce à des collaborations avec des compositeurs pour la génération [Truchet 2004] et avec Marc Chemillier pour l'analyse [Chemillier, Truchet 2003]. Cette phase a notamment permis d'établir une sorte de cahier des charges pour la conception d'un système de résolution de contraintes adapté à la musique. Pour l'analyse, trois problèmes sont issus de travaux de Marc Chemillier. Un projet concerne notamment la guitare. A partir d'une suite de chiffrage d'accords, il s'agit de trouver automatiquement une réalisation des accords et des doigtés pour la guitare. Les contraintes sont ici physiologiques, elles servent à caractériser les accords jouables pour le guitariste.

Pour toutes ces applications, nous avons déterminé que la méthode la mieux adaptée est une méthode innovante appelée recherche adaptative. Cet algorithme, dû à Philippe Codognet (Codognet, Diaz 2001), appartient à la famille des algorithmes de recherche locale, qui ont récemment largement prouvé leur efficacité sur des CSPs classiques. Il correspond particulièrement bien au cahier des charges musicales pour de nombreuses raisons, notamment la possibilité d'exploiter le processus de résolution « en cours de route » pour fournir des solutions approchées au problème.

La recherche adaptative a été intégrée dans OpenMusic sous la forme d'une librairie nommée OMClouds [Truchet, Assayag, Codognet 2001, 2003a, 2003b, 2003c]. Cette dernière permet de définir les contraintes visuellement dans un patch. Pour résoudre le problème, les contraintes sont d'abord automatiquement traduites en fonctions d'erreur, qui mesurent à quel point la contrainte est violée. Le résolveur tente

de trouver une solution en minimisant les erreurs. Cela permet de traiter naturellement les problèmes sans solutions, et plus généralement de donner facilement des solutions approchées au problème [Truchet, Codognet 2004]. Ces solutions approchées peuvent être éditées visuellement dans un objet OM et sont actualisées constamment au cours de la résolution [Truchet, Diaz, Codognet 2003] [Truchet, Agon, Assayag 2001] [Truchet, Agon, Assayag, Codognet 2001]

Plusieurs pièces musicales ont été réalisées avec les outils développés dans ce projet :

- *Empreinte sonore de la fondation Bayeler* de Georges Bloch, une pièce in situ articulée à la visite de l'exposition Abstraction/Ornement dans cette fondation à Bâle.
- *Coïncidences* pour orchestre de Fabien Lévy (Billaudot Editeur)
- *Aschenblume* pour ensemble de Mauro Lanza
- *Burger Time* pour tuba et électronique de Mauro Lanza
- *Droben* pour contrebasse, ensemble et électronique de Michael Jarrell, assistant Gilbert Nouno.

14.3. Programmation concurrente avec contraintes

Projet REACT, 2006-2008

En 2006 j'ai accueilli dans mon équipe Camilo Rueda, un collaborateur de longue date de l'Ircam actuellement directeur du département d'ingénierie informatique de l'université pontificale Javeriana de Cali (Colombie), comme chercheur invité. Cette collaboration faisait suite au projet Avispa sur lequel nos équipes avaient travaillé ensemble en 1996-1998.

Nos collaborations ont porté sur deux axes :

- Explorer l'utilisation des calculs temporels de processus de contraintes concurrents, en particulier le calcul NTCC (Diaz, Rueda, Valencia 1997), pour la modélisation de systèmes de génération musicale.
- Elaborer un projet conjoint de recherche entre l'Ircam, l'école polytechnique et l'Universidad Javeriana-Cali en Colombie, visant à trouver des formalismes issues de la théorie de processus de contraintes concurrents (ccp) assez généraux pour être applicables à la modélisation de systèmes complexes dans plusieurs domaines incluant la musique (projet REACT)

Les calculs des processus concurrents sont des formalismes mathématiques basés sur l'idée que pour comprendre le comportement d'un système d'agents (processus) il suffit d'observer ses interactions. Dans les calculs ccp l'interaction entre deux agents consiste à échanger de l'information partielle sur les valeurs des variables

communes. Dans NTCC, en particulier, les observations sont des couples de contraintes. Observer le couple (c,d) veut dire que quand l'information est c avant l'interaction alors elle est d après. Comme chaque contrainte peut représenter de l'information complexe ce qui est observé du système peut aussi être complexe. Tel est le cas, par exemple, des interactions qu'on peut observer entre les musiciens lors d'une séance d'improvisation musicale ou d'interaction en général.

Modéliser un système au moyen d'un calcul des processus a trois avantages. D'une part, le calcul étant formel, on peut prouver des propriétés du système modélisé avant de faire tourner son simulateur. D'autre part, le calcul étant modulaire, on peut raffiner de façon contrôlée le modèle du départ pour observer plus des détails de son comportement. Finalement, le calcul étant un formalisme computationnel, le modèle est lui-même un simulateur.

Le projet REACT (Robust theories for Emerging Applications in Concurrency Theory) a été rédigé et déposé au près de Colciencias, l'institut colombien de soutien à la recherche scientifique. Il a été accepté en avril 2007. Le projet compte avec le soutien budgétaire et technique de Colciencias, l'Universidad Javeriana-Cali, le labo LIX de l'Ecole Polytechnique et l'Ircam. Les équipes impliquées (incluant l'équipe Inria Comète) forment l'équipe de recherche associée de l'INRIA « Forces » dont le premier bilan a été dressé dernièrement [Aranda, Assayag & al 2009].

Quelles sont les limitations pratiques et théoriques des calculs ccp ? C'est la question qu'il prétend explorer en considérant l'utilisation des calculs dans trois domaines, la sécurité des protocoles d'authentification, la biologie moléculaire et l'interaction musicale.

Une représentation Ntcc du modèle « factor Oracle » (FO) pour l'improvisation musicale [Assayag, Dubnov 2004] a été construit [Rueda, Assayag, Dubnov 2006]. Le calcul ntcc permet d'exprimer le modèle FO sans une représentation explicite du graphe qui se déduit dynamiquement de l'information produite par les processus concurrents. Le modèle ntcc peut facilement intégrer des choix stochastiques pour la génération de variations. Une première version d'un interprète du calcul ntcc a été implémentée en Common Lisp.

Cette version utilise Screamer (Siskind & Allester 1993) comme moteur de contraintes et les processus natifs de MCL pour la concurrence. Le modèle FO a été implanté avec cet interprète, ainsi que le modèle de partitions virtuelles proposé par Myriam Desainte-Catherine [Allombert, Assayag, Desainte-Catherine, Rueda 2007]. Les résultats ont été encourageants, mais il est clair que la gestion de processus concurrents en MCL ne permet pas d'avoir l'efficacité requise dans des situations réelles d'interaction. Une implementation prototype de Gecode sous CommonLisp a été mise en œuvre en collaboration avec Antoine Allombert et Bruno Valèze. Maurizio

Toro Bermudez de l'Université de Cali a été reçu dans mon équipe en 2008 pour finaliser l'intégration OpenMusic / Gecode / NTCC [Toro Bermudez 2008].

Le projet REACT Plus qui fait suite à REACT est lancé en 2009.

15. Apprentissage et interaction

15.1. Interaction texte et musique

travail de DEA par Soufiane Bouyahi, étudiant à Paris VIII, en collaboration avec Peter Hanne et Jean-Pierre Balpe. 1998.

Ce projet a mené à la réalisation d'Ulysse, programme de génération textuelle, environnement qui se trouve à l'articulation de l'informatique musicale et de l'informatique linguistique. Il est basé sur les travaux de linguistique computationnelle de Jean-Pierre Balpe (Balpe 1997)

Il permet d'engendrer, de manière quasi-instantanée, un nombre indéfini de textes, obéissant à des règles syntaxiques, stylistiques et sémantiques, définies dans une base de données. Cette dernière contient des structures abstraites (des modèles de textes) ou littérales (des fragments pré-écrits) dont la recombinaison reconstitue un texte final dont la forme est contrôlée. Selon la manière dont est constituée la base données, tous les styles en sortie sont imaginables : poésie libre, avec métrique, texte de fiction, imitation d'un auteur, etc.

Ce système est conçu pour entrer en interaction avec un système musical, en temps différé ou en temps réel, avec une granularité temporelle qui est actuellement de l'ordre de la seconde. Il est couplé avec le pilote Midi du GRAME, MidiShare, et peut, grâce au codage en Midi, envoyer des informations sur les textes engendrés à un système musical comme PatchWork, OpenMusic, max ou FTS. Il peut aussi, en retour, recevoir de l'information provenant du système musical par le même canal. Ainsi, l'on peut envisager une interaction multimédia dans laquelle des algorithmes musicaux déterminent la morphologie du texte à générer, ou inversement, dans laquelle cette morphologie influe le cours du flux musical. Cette interaction peut, bien entendu, être étendue à d'autres médias comme la vidéo.

Techniquement, Ulysse a été implémenté en langage Java et est donc portable sur diverses plates-formes matérielles.

Ulysse a été conçu et développé en coordination avec un projet musical et poétique, « Trois mythologies et un poète aveugle » par Jean Pierre Balpe et Jacopo Baboni, créé à l'Ircam en 1998. La base de donnée existante est donc spécifique du style voulu par les auteurs pour ce spectacle. Il reste à définir d'autres bases données corres-

pendant à d'autres styles, ou plus généralement un éditeur permettant de construire et de tester très rapidement ces bases de données.

Ulysse est le résultat d'une collaboration entre mon équipe et le département Hypermédia de l'université Paris VIII. Il a donné lieu au co-encadrement d'un travail de DEA par Soufiane Bouyahi, étudiant à Paris VIII. Le modèle pour la génération linguistique a été conçu par Jean-Pierre Balpe.

15.2. Apprentissage du style musical pour l'interaction

Ce projet que j'ai initié dès 97-98 à l'Ircam en collaboration avec Shlomo Dubnov a donné lieu à plusieurs stages de DEA de 2000 à 2004. Il constitue la base théorique sur laquelle sera construite ensuite le projet OMax (Interaction Symbolique Improvisée) qui est détaillé plus loin.

Modélisation du style musical par apprentissage statistique : une application de la théorie de l'information à la musique. Olivier Lartillot, DEA ATIAM, Ircam - Université Paris VI, 2000.

Simulations d'improvisations à l'aide d'un automate de facteurs et validation expérimentale. Emilie Poirson. DEA ATIAM, Paris 6, juillet 2002, en collaboration avec Emmanuel Bigand (LEAD, U. Bourgogne, UMR 5022).

Apprentissage du style musical et interaction sur deux échelles temporelles, Nicolas DURAND, DEA ATIAM, Université Pierre et Marie Curie, PARIS VI, 2003

Attributs multiples dans un improvisateur automatique, Cyril Laurier, DEA ATIAM, UPMC Paris 6, 2004

Les théories musicales prédictives sont importantes pour capturer et représenter certains aspects du style musical qu'il est plus difficile de manipuler avec les systèmes à bases de règles. Ces théories supposent que la cognition musicale est guidée par une certaine « attente » de la part de l'auditeur. Cette attente se référerait à des ensembles de séquences-type (patterns) et de formes temporelles déjà connues, et susceptibles d'être étendus par de nouvelles séquences au fur et à mesure de l'apprentissage. D'un point de vue plus formel, ces théories ont été souvent liées aux modèles stochastiques qui estiment la probabilité pour que certains éléments musicaux apparaissent dans un contexte musical donné. Les exemples les plus connus sont les chaînes de Markov déjà utilisées intensivement dans la musique par ordinateur. Le problème principal avec ces modèles est qu'il est difficile d'estimer leurs paramètres et qu'ils augmentent exponentiellement en coût avec la taille du contexte.

Mon approche a consisté à montrer avec Shlomo Dubnov que les algorithmes de compression universels, bien connus en traitement des données, fournissent un cadre général, efficace et flexible au problème de l'apprentissage automatique (machine learning) dans le domaine du style musical [Dubnov, Assayag, El Yaniv 1998]

[Assayag, Dubnov, Delerue 1999]. Par apprentissage, nous entendons l'analyse et le codage efficace de caractéristiques stylistiques importantes d'un flux musical, de sorte que des opérations musicales significatives puissent être réalisées sur les représentations résultantes. Par exemple, nous voulons que l'ordinateur puisse restituer dans un contexte créatif les caractéristiques apprises, notamment dans l'improvisation ou dans l'aide à la composition.

Les algorithmes de compression universels font des hypothèses minimales sur les sources stochastiques sous-jacentes des séquences analysées. Ainsi peuvent-ils être utilisés dans une grande variété de situations musicales et stylistiques. Ceci diffère des systèmes à base de règles ou de connaissances (knowledge based systems) (Balaban & al 1992) dans lesquels un grand nombre d'informations spécifiques doivent être encodées sous la forme de descripteurs stylistiques et de règle de recombinaison.

Nous avons exploré des méthodes de compression qui calculent un dictionnaire des « phrases » rencontrées dans le déroulement du flux musical. Ces phrases, stockées dans un dictionnaire, peuvent ensuite constituer des contextes, à partir desquels sont estimées les « continuations » possibles du discours musical. Dans la phase de génération (re-synthèse), la partie finale de la séquence déjà générée est considérée comme un contexte, et le dictionnaire est utilisé pour lui trouver une continuation. Puis le procédé est réitéré sur la nouvelle séquence ainsi formée.

Ces expériences ont montré en particulier que la méthode de compression dite « Lempel-Ziv » (Ziv & Lempel 1977), semble fournir des contextes et des continuations produisant une génération raisonnablement proche de l'attente naturelle, et cela avec une grande efficacité.

Une des difficultés rencontrées a été liée au choix d'une représentation musicale universelle, linéaire, et préservant la généralité des algorithmes de compactage. Nous avons montré que la représentation MIDI n'est pas adéquate, excepté dans le cas le plus simple de la monodie. Nous avons donc utilisé une représentation spéciale pour la musique polyphonique, dans laquelle la hauteur, la voix, et le temps sont encodés dans des séquences linéaires de « super-symboles ». Pour cela, nous avons utilisé le vocabulaire des « langages de superposition » décrit par Marc Chemillier dans sa thèse sur les structures algébriques en musique (Chemillier 1989). Nous découpons le temps en tranches verticales correspondant à l'arrivée d'un nouvel élément dans la polyphonie. Les notes dont la durée occupe plusieurs tranches verticales sont décomposées en un symbole de début et une série de symboles de continuation (AA'A'A'A'). Les durées sont quantifiées et ramenées à un petit répertoire de valeurs possibles. Ainsi le discours polyphonique peut-il être linéarisé sous la forme d'une chaîne de symboles sur un alphabet produit. Chaque symbole est de la forme : $(p_1, p_2, \dots, p_n, D)$ dans lequel D est la durée de la tranche verticale et p_1 à p_n sont les N notes superposées dans cette tranche. Les résultats de la méthode LZ appliquée à

ces séquences ont été tout à fait satisfaisant du point de vue de l'attente stylistique et de la capacité à engendrer des variantes crédibles. Les tests ont été menés sur deux types de corpus : les inventions à deux voix de J.S. Bach et des standards de Jazz de Charlie Parker et Thelonious Monk.

Cette recherche a ouvert des perspectives tout à fait intéressantes sur la possibilité d'une nouvelle approche de la CAO : en demandant à l'ordinateur de faire l'apprentissage de ses propres matériaux musicaux, puis de proposer des variantes, recombinaisons et développements, le compositeur peut explorer en profondeur les conséquences de ses choix musicaux sans avoir besoin de les formaliser complètement.

Ce travail a d'autre part suscité un certain nombre d'interrogations :

- étude des relations entre les modèles théoriques pour la compression des données (markov, LZ, etc.) et les représentations cognitives, basée sur l'idée que « compresser c'est découvrir des structures, c'est comprendre »
- essayer de comprendre pourquoi le modèle LZ semble marcher mieux en génération que les modèles à base de Markov. En quoi correspond-il mieux à notre attente musicale ? En quoi les surprises qu'il génère sont-elles acceptables comme variations et non comme cassures ?
- applications temps réel à l'improvisation
- applications à la classification automatique et à la recherche intelligente dans les bases de données par mesure de l' "information mutuelle" entre paires de séquences.

Dans le cadre de son DEA [Lartillot 2000], Olivier Lartillot a repris le travail commencé auparavant par S. Dubnov et moi-même sur la modélisation du style musical par apprentissage statistique. Cette technique est liée aux modèles stochastiques qui estiment la probabilité pour que certains éléments musicaux apparaissent dans un contexte musical donné.

Des méthodes de compression universelles (telles la méthode Lempel-Ziv) calculent un dictionnaire des « phrases » rencontrées dans le déroulement du flux musical. Ces phrases, stockées dans un dictionnaire, peuvent ensuite constituer des contextes, à partir desquels sont estimées les « continuations » possibles du discours. Dans la phase de génération (re-synthèse), la partie finale de la séquence déjà générée est considérée comme un contexte, et le dictionnaire est utilisé pour lui trouver une continuation. Puis le procédé est réitéré sur la nouvelle séquence ainsi formée.

O. Lartillot a pu améliorer de manière très sensible l'algorithme initial en imaginant un ensemble de raffinements agissant directement sur la qualité de la sortie musicale :

- détection et inhibition des « attracteurs », structures circulaires du dictionnaire qui provoquent un bouclage infini (trille)

- séparation des informations d'analyse et de synthèse. Par ce procédé, l'utilisateur peut déterminer quels paramètres musicaux sont à utiliser dans l'analyse (par exemple, l'analyse peut être effectuée sur les hauteurs uniquement). Les autres paramètres (paramètres de synthèse) sont captés par l'analyse et stockés en attente de la re-synthèse. Ainsi, l'analyse, portant sur des classes plutôt que des éléments singuliers est-elle plus puissante, le rendu de synthèse restant réaliste.
- Mécanismes de « backtracking » (retour arrière) permettant de remettre en question un choix de synthèse s'il s'avère conduire à une impasse.
- Spécification de contraintes musicales dans la phase de re-synthèse

Emilie Poirson [Poirson 2002] pour sa part a abordé le projet avec un nouveau modèle, l'Oracle des facteurs (Allauzen, Crochemore & Rafinot 1999). J'ai montré [Assayag, Dubnov 2004] que ce dispositif formel issu de la recherche en stringology et appliqué à la reconnaissance de sites sur les chaînes macromoléculaires comme l'ADN était équivalent à un arbre de suffixes collapsé le long de sa branche principale. De ce fait, moyennant une certaine perte d'information (qui en fait un compresseur avec pertes, lossy compressor) cette structure donne accès à tous les contextes du dictionnaire avec une efficacité linéaire en temps et en espace.

Après avoir examiné divers scénarios d'apprentissage non supervisé : par compression de données (modèle LZ), par arbres de suffixes prédictifs avec le modèle PST [Assayag, Dubnov, Lartillot, Bejerano 2003] l'Oracle des facteurs a fait l'objet de recherches intensives. Le résultat sur la structure implicite d'arbre des suffixes nous a permis d'établir que tous les modèles étudiés précédemment étaient en réalité des instances d'arbres de suffixes dans lesquels on tolérait une perte plus ou moins grande d'information. Cette perte d'information provoque des comportements en « dents de scie » des contextes (la mémoire à court terme à partir de laquelle est décidée la génération du prochain événement), ce qui permet de ramener tous ces modèles à des modèles de Markov à contexte variable. Cette variation des contextes est essentielle au concept d'improvisation : recombinaison de motifs puisés dans la mémoire du musicien, mais avec une fonction d'oubli qui empêche la pure et simple reproduction du passé.

Des expériences ont été menées par Emilie Poirson et moi-même avec des pianistes de Jazz à qui il a été demandé d'improviser sur des standards donnés. Le matériau a été enregistré en audio et en Midi, puis soumis à l'algorithme d'apprentissage. Des tests perceptifs (en collaboration avec Emmanuel Bigand de l'U. de Bourgogne) sur des sujets de compétence moyenne ont été proposés : il s'agissait de tests d'écoute « aveugles » sur des séquences issues du jeu humain et du jeu machinique. Les sujets devaient positionner en continu un point dans un espace à 2 dimensions dont les extrêmes étaient : (Instrumentiste1 <-> Instrumentiste2) et (Clone numérique de 1 <-> Clone numérique de 2).

Il en est ressorti que la discrimination entre le style 1 et 2 était quelque peu gommée dans le cas des clones. La discrimination humain / clone, si elle finissait bien sûr par être établie, nécessitait néanmoins une durée étonnamment stable (de l'ordre de 30 secondes) qui semble correspondre à la durée moyenne de la grille.

Enfin, des tests plus informatiques ont été faits sur les fonctions de comparaison permettant d'établir la redondance entre les blocs musicaux considérés comme atomiques : égalité stricte, similarité texturale, similarité harmonique. La similarité texturale a donné des résultats très intéressants en termes créatifs, elle sera reprise dans le projet OMax détaillé plus loin.

Dès les prémices de ce projet sur la simulation stylistique, Olivier Delerue a réalisé dans mon équipe en 1998 un premier prototype en Java permettant l'interaction en temps-réel et préfigurant le futur système OMax. Nous en donnons ici quelques éléments.

Le logiciel utilise directement les résultats d'analyse générés sous OpenMusic (création de paires de fichiers : lexique et arbre de continuation) et calcule en temps réel des continuations à partir d'un symbole initial choisi aléatoirement. Le résultat sonore est perçu via un synthétiseur. L'aspect temps-réel de cette implémentation a en outre permis la réalisation d'un accompagnateur automatique fondé sur le même principe. Les notes jouées par un interprète sur un clavier midi déclenchent la recherche dans l'arbre d'un mot (un accord de deux notes) constituant une continuation possible de ce qui a été joué précédemment. Parmi ces continuations possibles, ne sont gardées que celles dont la note de soprano coïncide avec celle émise par l'interprète. Toujours suivant la même loi de pondération, une de ces continuations est choisie aléatoirement de manière à générer une note de basse. A chaque itération, le préfixe courant est augmenté de la dernière continuation choisie. L'algorithme temps-réel d'exploration est rendu possible par une structuration de l'arbre optimisée : renversement des chaînes de préfixes, organisation par mot, par note de soprano ou de basse.

Ce prototype réalisé par Delerue a été la première réalisation temps-réel interactive des modèles d'apprentissage statistiques de séquences musicales que j'ai proposés. Il permettait, de manière alors inédite, de jouer en contrepoint avec l'ordinateur, après apprentissage d'une séquence polyphonique (en l'occurrence les expériences ont été menées avec le corpus complet des inventions à deux voix de J.S. Bach).

Il faut mentionner aussi le travail de Cyril Laurier, sur la notion d'attributs multiples [Laurier 2004] : le sujet en est l'étude d'un modèle statistique, les Multi-attribute Prediction Suffix Graph ou MPSG (Triviño-Rodriguez & Morales-Bueno 2001), pour l'apprentissage du style et l'improvisation. L'objectif de ce stage de recherche était d'implémenter les MPSG et d'en vérifier les propriétés puis de voir s'ils pouvaient constituer une alternative crédible aux modèles explorés dans l'équipe. Nous avons

vu qu'au delà de l'utilisation monodique faite dans l'article de référence nous pouvions employer le système sur des polyphonies, moyennant une représentation adaptée. Nous avons fait une étude des attributs plus riches musicalement que simplement les hauteurs et les durées des notes. En effet, avec des paramètres de plus haut niveau tels que l'harmonie ou une topologie de motifs nous pouvions envisager une analyse plus macroscopique. Ensuite nous avons comparé qualitativement les différents modèles statistiques connus, et analysé empiriquement la complexité du modèle. Le bilan est que l'extension à un système multi-attributs dans lequel on veut pouvoir choisir dynamiquement la priorité donnée à tel ou tel paramètre musical dans la conduite de la génération entraîne une complexification considérable des modèles simples utilisés jusque lors. Les coûts en calcul rendent le temps-réel impossible et les résultats musicaux ne sont pas à ce stade convaincants. Nous avons mis à jour un principe d'indétermination : lorsque les lexèmes musicaux sont utilisés comme des atomes multi-paramétriques (le cas avec LZ et avec Oracle) les modèles sont simples, efficaces et convaincants musicalement. Mais ces modèles ne sont pas de nature à inventer des lexèmes qui n'ont pas été rencontrés dans l'apprentissage. Lorsque l'on emploie des systèmes à multiples points de vue (Conklin & Witten 1995) ou multi-attributs, le système peut inventer de nouveaux lexèmes musicaux (par exemple de nouvelles combinaisons hauteur / durée / intensité) mais au prix d'une complexité difficile à gérer et d'une perte de contrôle musical (les nouveaux lexèmes créent des ruptures gênantes dans le flux engendré). La conclusion de cette étude est qu'il faut conserver l'approche par lexèmes atomiques mais augmenter la base d'apprentissage de manière à « rencontrer » le maximum d'unités musicales significatives dans l'apprentissage.

Une autre approche non atomique a cependant été tentée dans le cadre du stage de DEA de Nicolas Durand [Durand 2003]. Il s'agissait de tenter de résoudre la question des points de vue non pas par des objets multi-attributs mais par des oracles parallèles opérant sur des attributs différents (hauteurs, durées) et de tenter le recollement à la génération. Les résultats ont montré que les oracles indépendants engendraient des séquences convaincantes pour les paramètres considérés, mais que les procédures de recollement souffraient des mêmes défauts que pour les MPSPG. La conclusion générale de cette série d'étude est que les meilleurs résultats sont obtenus par des modèles de séquences linéaires opérant sur un alphabet qui est le produit cartésien des domaines des paramètres musicaux prégnants et sur des bases d'apprentissage de grande ampleur. C'est la méthode qui sera employée dans OMax.

15.3. Partitions interactives

Thèse d'Antoine Allombert, Université de Bordeaux en co-tutelle Labri / Ircam (2006-2009)

Le but de ces travaux en co-direction avec Myriam Desainte-Catherine est de dégager un modèle de partitions numériques permettant l'interprétation de pièces musicales au travers de « points d'interaction » introduits par le compositeur. Les partitions sont composées de structures temporelles pouvant représenter divers objets musicaux : processus de synthèse, sons... et d'événements définis comme interactifs. Ces événements sont dits interactifs car le moment de leur déclenchement est choisi par l'interprète pendant l'exécution. Le compositeur a également la possibilité de lier les structures par des contraintes temporelles basées sur les relations de Allen pour faire apparaître une cohérence générale de sa pièce et ainsi définir et limiter un espace de libertés dans lequel l'interprète va pouvoir s'exprimer. Il y a donc deux phases successives : la composition au cours de laquelle le compositeur va définir les structures, les lier au moyen des contraintes temporelles et placer des points d'interactions et l'exécution pendant laquelle l'interprète va jouer la pièce en déclenchant les événements interactifs le système ayant la charge du déclenchement des événements non interactifs (statiques).

Notre objectif était de faire émerger un modèle de partitions pour la phase de composition ainsi qu'un outil pour les éditer qui permettent de définir chacun de leurs éléments et de maintenir les contraintes temporelles pendant l'édition ainsi qu'un système d'exécution des pièces qui offrent la possibilité à l'interprète de déclencher les événements interactifs tout en maintenant les contraintes temporelles.

Le modèle théorique des partitions étant bien défini, nos efforts se sont portés sur son intégration dans OpenMusic, logiciel de composition assistée par ordinateur développé au sein de l'équipe, et plus particulièrement en s'appuyant sur le système des Maquettes d'OpenMusic qui se prête bien à notre modèle de partition. Cette intégration s'est faite en utilisant la librairie de contraintes Gecode (Tack 2009) pour permettre le maintien des contraintes pendant l'édition. Cette phase de développement effectuée, nous avons envisagé les solutions possibles pour le système d'exécution, nous avons ainsi retenu deux possibilités : un modèle basé sur les réseaux de Pétri et un autre utilisant le langage de contraintes concurrentes NTCC (Non deterministic Temporal Concurrent Constraint Calculus) (Mogens & Valencia 2002). L'intégration du modèle de partitions à OpenMusic et les réflexions autour des choix de modèle pour l'exécution ont donné lieu à la publication d'un article lors de la conférence Sound and Music Computing 2006 en collaboration avec Camilo Rueda de l'université de Javeriana (Colombie) et Myriam Desainte-Catherine de l'université Bordeaux 1 [Allombert, Assayag, Desainte-Catherine, Rueda 2006]. Afin de comparer l'efficacité des deux modèles, nous avons engagé leur implémentation toujours au travers d'OpenMusic. Une première implémentation du modèle utilisant les ré-

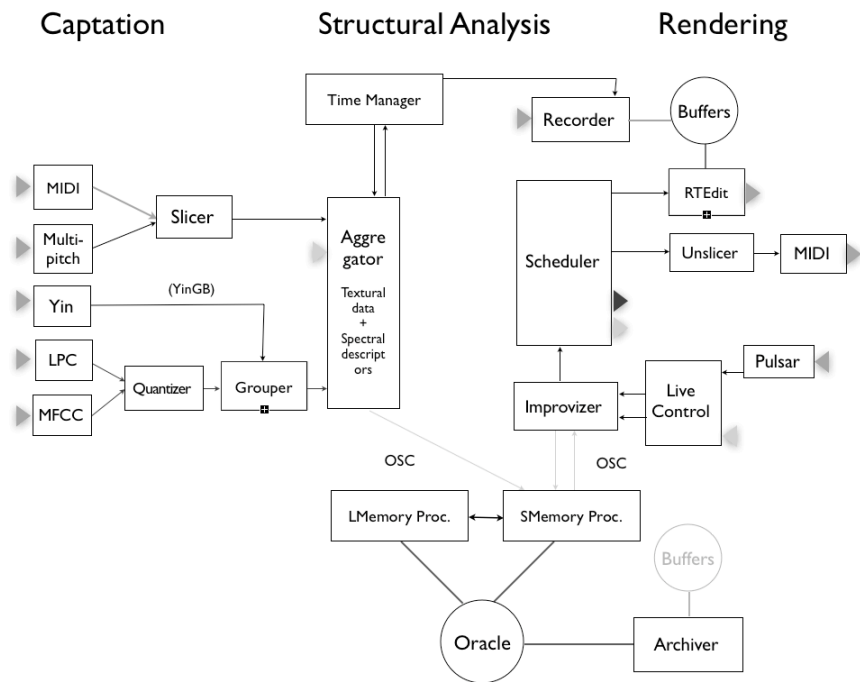
seaux de Pétri a ainsi été effectuée permettant d'entamer des tests. Cependant, cette implémentation reste incomplète et nécessite encore quelques développements pour être directement utilisable par des compositeurs et des musiciens. Le modèle basé sur NTCC n'a pas encore donné lieu à une implémentation, mais l'un des éléments principaux de ce modèle (un magasin de contraintes) a été ajouté au modèle initial basé sur les réseaux de Pétri. Ce nouvel élément également implémenté dans la version OpenMusic du modèle permet la définition de contraintes globales en plus des contraintes temporelles entre les structures. Le compositeur peut ensuite définir des stratégies à adopter au cas où au cours de l'exécution l'une des contraintes globales seraient violées : soit ne pas jouer la structure dont le déclenchement viole une contrainte, soit repousser le déclenchement de cette structure jusqu'à ce qu'il devienne possible sans violation de contrainte ou de modifier les paramètres de la structure pour que son déclenchement soit compatible avec l'ensemble des contraintes contenues dans le magasin [Allombert, Desainte-Catherine, Assayag 2008].

Ce projet se prolonge actuellement dans le cadre d'un financement ANR (projet de plate-forme Virage).

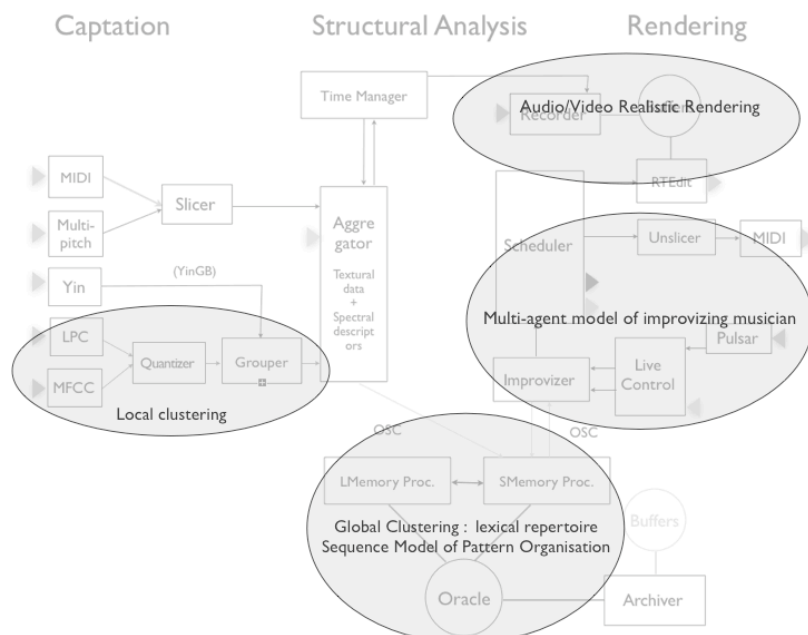
15.4. OMax : modélisation du style, de l'interaction, de l'improvisation

En collaboration avec les OMax Brothers (Georges Bloch, Marc Chemillier et Shlomo Dubnov) j'ai conçu plusieurs modèles théoriques et réalisé divers prototypes logiciels dans le domaine de la modélisation du style et de l'improvisation. Ces recherches ont évolué vers l'interaction en temps réel avec le système OMax. J'ai pu accueillir Marc Chemillier et Georges Bloch en délégation CNRS (2005-2006 et 2006-2007) pour m'accompagner sur ce projet. OMax marque une étape nouvelle en matière d'interaction (on a pu parler à ce sujet d'*interaction symbolique* ou *interaction cognitive*) dans la mesure où il ne s'agit plus de traiter le signal entrant selon des effets numériques continus ou des déclenchements univoques, mais de modéliser la mémoire à long terme de la performance et son style jusqu'à élaborer un *clone instrumental*.

Si des travaux sur la mémoire d'improvisation ont déjà eu lieu (Ramallo, Rolland, Ganascia 1999) ainsi que des systèmes interactifs de continuation du jeu instrumental (Pachet 2003), OMax est le premier système capable d'élaborer en continu un clone du musicien en train de jouer, cela aussi bien du point de vue symbolique (la logique d'enchaînement des motifs) que sonore (la restitution crédible des articulations instrumentales avec le son d'origine). Il est alors loisible de proposer une interaction complète dans laquelle le musicien élabore une improvisation en dialogue ou en contrepoint avec son clone. J'ai proposé pour cette situation inédite la formule : *réinjection stylistique* [Assayag, Bloch, Chemillier 2006].



Vue diagrammatique d'OMax



Vue fonctionnelle d'OMax

15.4.1. OMax version 1

Tout d'abord les travaux sur la modélisation du style, menés en collaboration avec Shlomo Dubnov, ont permis de mettre au point, puis d'évaluer plusieurs modèles d'apprentissage permettant de construire automatiquement une représentation riche dérivée d'une séquence musicale de référence. Ces modèles sont principalement ceux issus de la compression de données (algorithme Incrémental Parsing issu de la compression Lempel-Ziv), des arbres de suffixes prédictifs (PST), et plus récemment, d'une structure proposée par M. Crochemore et adaptée par nos soins aux problématiques musicales, l'Oracle des Facteurs (FO).

Marc Chemillier a d'abord étudié les conséquences opérationnelles et formelles des travaux de M. Steedman sur la modélisation des mécanismes de substitution harmonique dans le jazz à l'aide de grammaires (Steedman 1984) [Chemillier 2004]. La notion de réécriture, qui consiste à remplacer un segment par un autre dans une séquence, modélise la notion musicale de variation. En implémentant un tel modèle, on peut explorer automatiquement des niveaux très profonds de variation, qui dépassent ceux qu'un improvisateur humain contrôle lorsqu'il joue. Le paramétrage sensible du modèle, en vue d'une utilisation en temps réel, conduit à des problèmes théoriques intéressants sur la forme des séquences générées par la grammaire de Steedman. Jean-Brice Godet a effectué son stage de DEA ATIAM sur ce thème.

Ces deux premiers axes de recherche mettent en œuvre des concepts issus de la théorie des langages (automates pour l'apprentissage, grammaires pour les substitutions harmoniques). Leurs fondements théoriques ont été exposés dans deux articles d'un numéro spécial de la revue *Soft Computing* coordonné par G. Assayag, V. Cafagna et M. Chemillier, paru en octobre 2004, intitulé « Formal Systems and Music » [Assayag, Cafagna, Chemillier 2004].

La troisième ligne est constituée par la réflexion autour d'une architecture permettant à la fois une production et une interaction sonore temps-réel, et une « réflexion » permettant d'élaborer des stratégies et des décisions musicales à plus long terme, voire de reconfigurer le schéma de composition en fonction de l'interaction. Cette idée a été exposée notamment sous le titre « articulation CAO – temps réel ».

A partir de ces lignes de forces, l'objectif général a été la réalisation d'un environnement de conception et d'expérimentation, donnant à son utilisateur un accès flexible à :

- des modèles d'apprentissages compatibles avec le temps réel, permettant de capter une performance humaine et de mettre à jour de manière incrémentale une représentation riche.
- des modules de captation permettant de gérer du Midi comme de l'audio

- des modules génératifs, basés soit sur la représentation résultant de l'apprentissage, soit sur des modèles formels *a priori* (grammaires de Steedman, mais aussi toute procédure de CAO déjà disponible dans le référentiel d'OpenMusic)
- des modèles de contrôle gérant les rapports d'échelles temporelles entre la captation et la génération temps-réel d'une part et le niveau « réflexif », analytique et génératif, d'autre part.

Nous avons élaboré un prototype d'un tel environnement, baptisé OMax. Comme son nom l'indique il s'agit d'une forme de connexion des programmes OpenMusic et Max, dotée de l'appareil algorithmique permettant d'expérimenter les trois points susmentionnés, en particulier :

- un protocole de communication entre Max et OpenMusic permettant de véhiculer des informations de bas niveau (événements Midi, références vers des événements audio) et de haut niveau (labels harmoniques)
- une architecture côté Max, basée sur un ensemble d'entrées Midi et Audio, de séquenceurs Midi et Audio, fonctionnant soit en temps pulsé, soit en temps libre, et un ensemble d'encodeurs/décodeurs gérant différents niveaux d'information.
- une architecture côté OpenMusic, basée sur une classe complexe, nommée « Orchestre », dans laquelle chacun des musiciens virtuels peut instancier toute la technologie d'apprentissage et de génération et garder à jour une représentation du temps (temps local et global). Cet orchestre peut « réfléchir » analytiquement sur tout le passé d'une session jusqu'au temps courant, mais peut aussi puiser dans des sessions archivées, voire dans un répertoire appris.

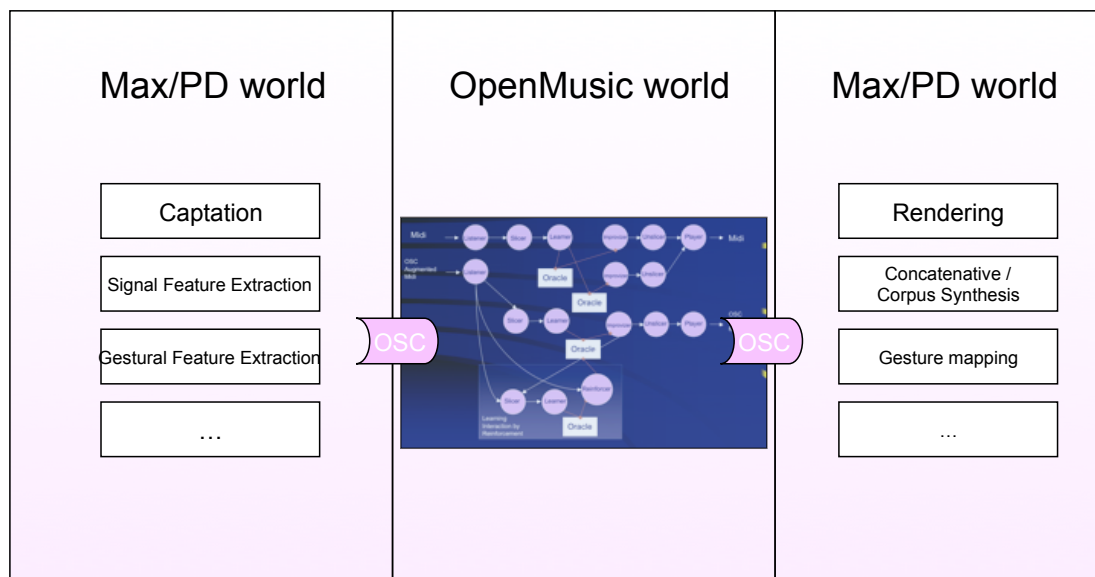
Nous avons ensuite mis en place une série d'expérimentations avec des musiciens, notamment Bernard Lubat (projet « Médiation scientifique sur Internet » financé par le Ministère de la recherche), et Jean-Rémy Guédon, qui ont permis un travail fin de mise au point des paramètres du système. La collaboration avec Jean-Rémy Guédon a abouti à un concert de l'orchestre National de Jazz à l'ircam en 2003. Ces expérimentations ont aussi permis d'archiver une grande quantité de données (données Midi/Audio, base de connaissances issues de l'apprentissage). Enfin, elles ouvrent la voie à une exploration méthodique des phénomènes d'interaction en œuvre dans les situations d'improvisation collective, aussi bien dans leurs aspects techniques (musicaux, informatiques), qu'anthropologiques.

De manière à faire le point sur l'état de l'art et ouvrir le dialogue avec les musiciens et chercheurs œuvrant dans des domaines similaires, nous avons organisé une grande manifestation dans le cadre de Résonances 2004, le « Workshop on Improvisation and Computer » qui a permis un échange fructueux avec des personnalités telles que Robert Rowe, Steve Coleman, Bernard Lubat, Roscoe Mitchell, Joel Cha-

dabe, David Wessel, Benny Sluchin, etc., et une confrontation entre des points de vue issus d'esthétiques très variées.

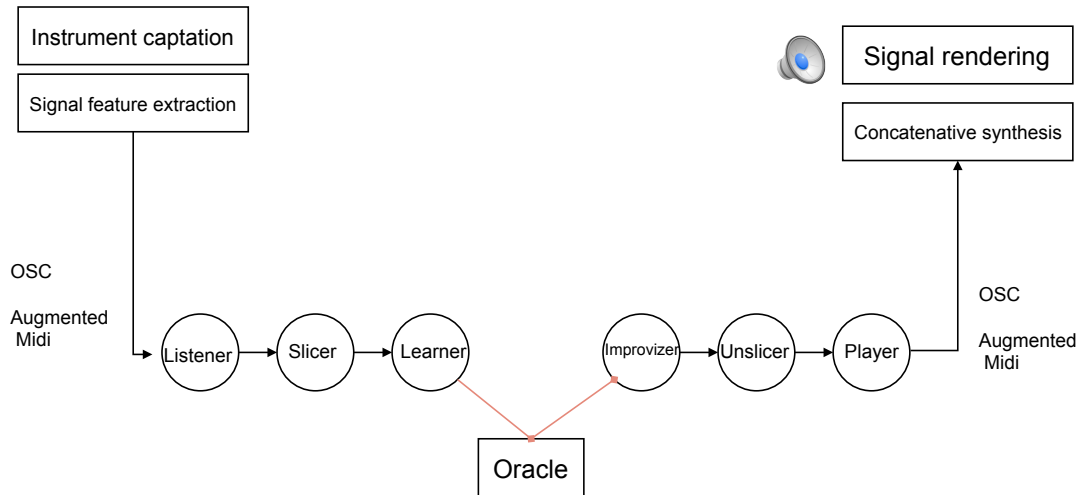
Ces diverses expériences ont permis d'aborder des situations différentes et de mettre à l'épreuve la flexibilité d'OMax :

- échange entre l'ordinateur et un instrument Midi
- échange entre l'ordinateur et un instrument acoustique (dans ce cas, les procédures de modélisation opèrent sur une représentation Midi obtenue à l'aide du module Yin~ (de Cheveigné & Kawahara 2002) et gardent des pointeurs vers les enregistrements audio.
- échange d'improvisations médiatisées par l'apprentissage, en temps libre ou pulsé, en contexte harmonique libre ou contraint.
- déclenchement de procédures compositionnelles de large portée temporelle dans OpenMusic en réponse à une proposition du musicien humain
- situation mixte de pièce composée incorporant des parties ouvertes avec échange musicien / ordinateur (*Tribute to Monk* de G. Bloch avec P. Leclerc au saxophone)



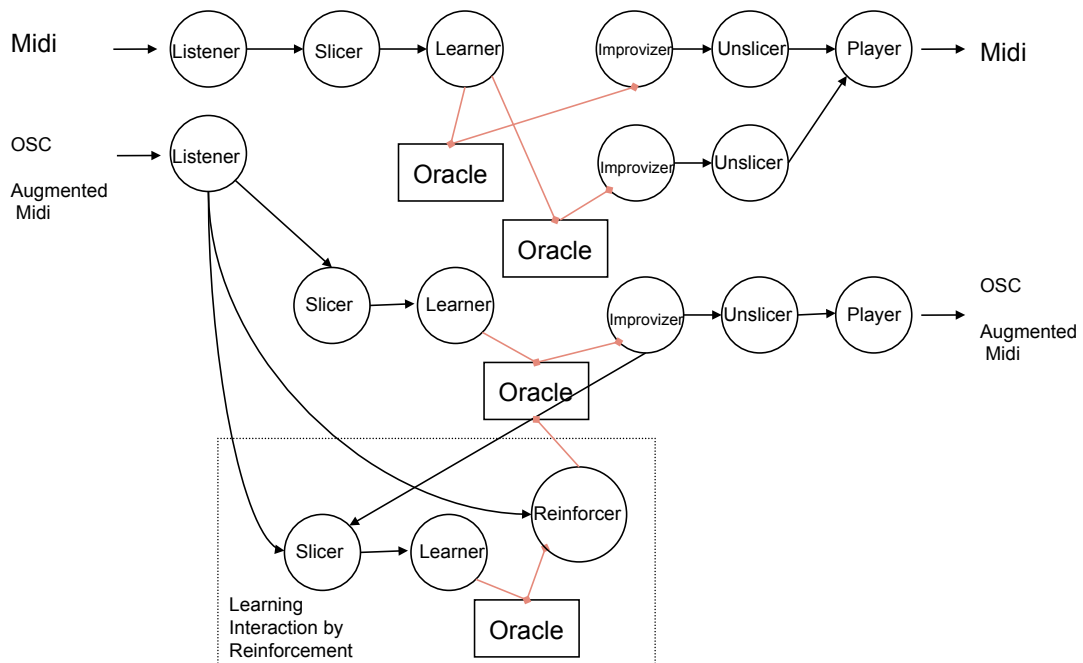
Vue architecturale d'OMax

Concurrent Improvising Agents Topology

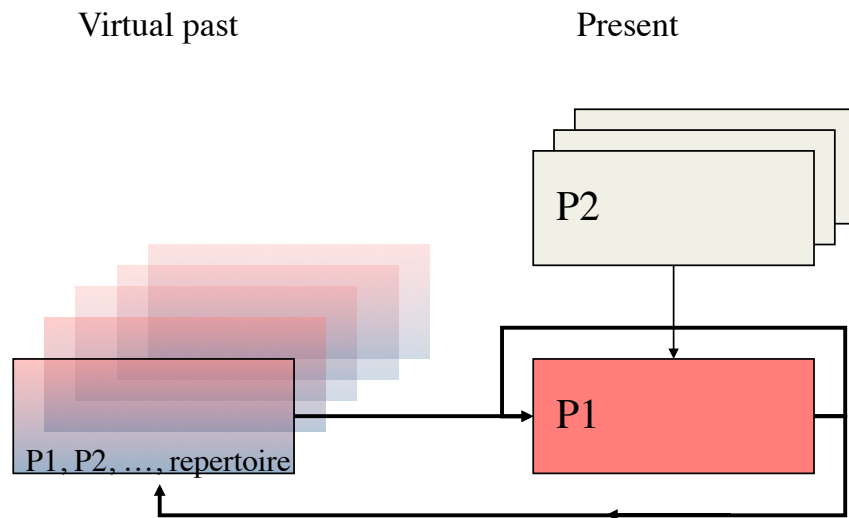


OMax : exemple de topologie élémentaire d'apprentissage et d'improvisation dans l'univers OpenMusic

Concurrent Improvising Agents Topology



OMax : exemple de topologie complexe d'apprentissage et d'improvisation dans l'univers OpenMusic



Improvisation as Stylistic Injection / Reinjection

Omax : le schéma de la réinjection stylistique

15.4.2. OMax version 2

Omax a été entièrement réécrit en 2006, le premier prototype ayant atteint ses limites. Du côté OpenMusic, l'implémentation monolithique initiale a fait place à une assemblée d'agents concurrents [Assayag, Bloch, Chemillier & al 2006] appartenant à cinq classes :

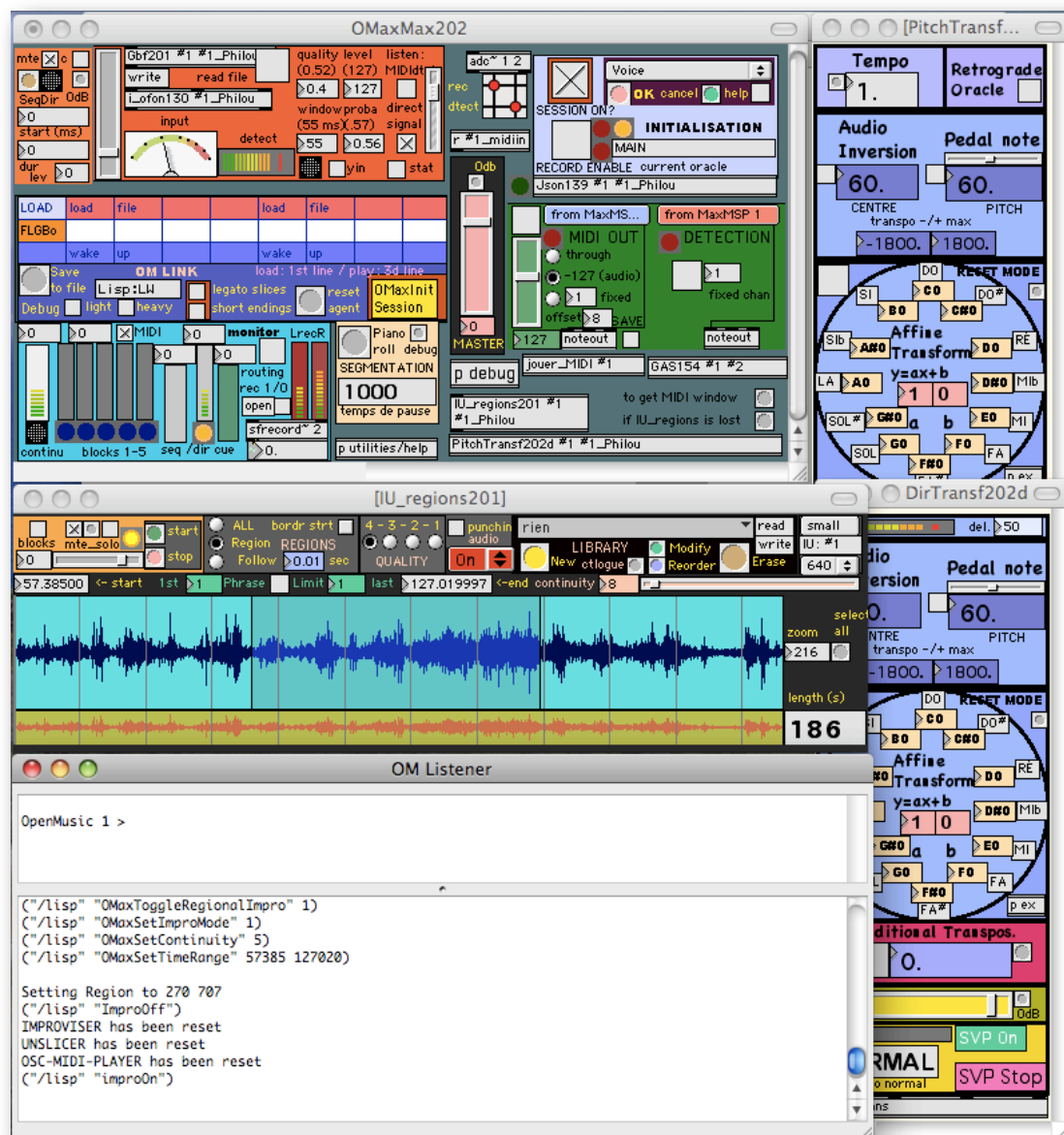
- *Listener* écoute les flux de descripteurs d'événements musicaux en entrée
- *Slicer* construit la représentation de tranches polyphoniques constituant le vocabulaire formel (*lexèmes musicaux*)
- *Learner* implémente le modèle d'apprentissage (l'Oracle des Facteurs dans la version courante)
- *Improvizer* joue des séquences improvisées
- *Unslicer* décode les séquences improvisées pour retrouver, à partir des tranches polyphoniques, un flux de descripteurs d'événements musicaux
- *Player* joue le flux en sortie

Ces agents fonctionnent en parallèle et en continu pendant l'interaction, et peuvent être instanciés en autant d'individus que souhaité, connectés selon des topologies diverses, pour réaliser des situations complexes de jeu.

Le modèle d'apprentissage lui-même a été considérablement amélioré par la définition d'une construction théorique que j'ai proposée, les Arbres de Liens Suffixes (ALS). Ces éléments, présents dans l'Oracle mais négligés par les auteurs de l'algorithme initial (car cet algorithme n'était pas destiné initialement à la génération), ont été théorisés et permettent d'atteindre de manière optimale toutes les zones de la mémoire musicale [Assayag, Bloch 2007]. Une seconde amélioration importante a consisté en la définition d'un critère de qualité dans les choix de recombinaison, prenant en compte la cohérence rythmique à long terme qui permet d'identifier des contextes où les rythmes ne sont pas identiques mais similaires du point de vue de leur densité. Ces deux facteurs ont beaucoup apporté en termes de qualité musicale et de contrôle par l'utilisateur.

Du côté de l'interface de jeu en Max, la principale innovation est l'accès, donné à l'utilisateur, à une représentation de la mémoire de jeu, l'Oracle comportant une représentation linéaire du temps de la session courante ou des sessions archivées. Il est ainsi possible de piloter l'improvisation synthétique vers une région plutôt qu'une autre, et un mécanisme d'archivage de régions d'intérêt a été rajouté. En particulier, une région appelée « poursuite » se cale en permanence entre t_0 (présent en temps réel) et $t_0 - k$, fournissant ainsi une variation continue des événements récents, ce qui permet à l'instrumentiste de véritablement piloter le système par son propre jeu.

Enfin des transformations formelles sont maintenant proposées (rétrogradation, inversion, transformations affines, transformations de tempo).

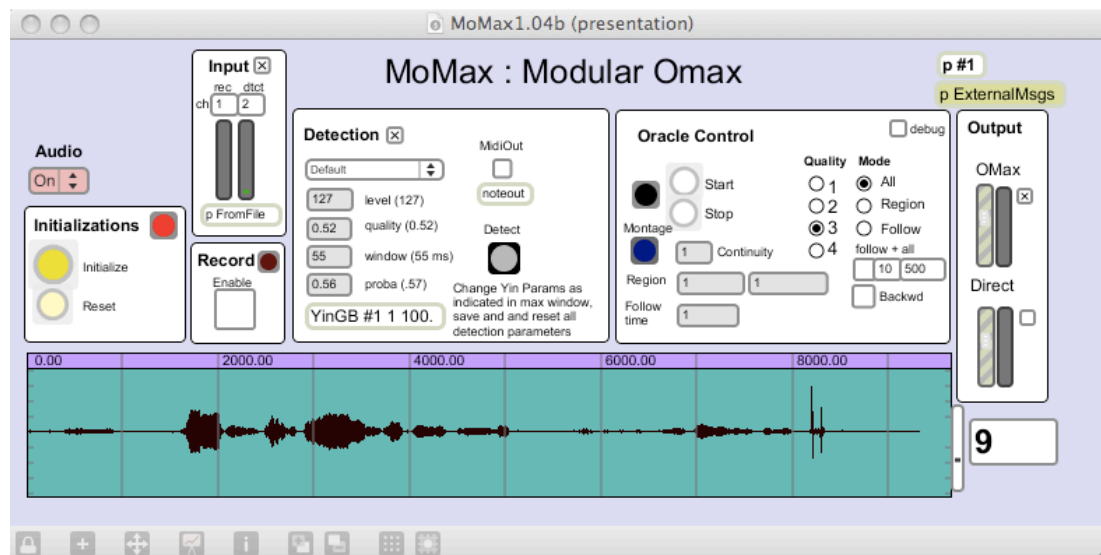


L'interface de contrôle d'OMax en Max, conçue par Georges Bloch. En bas, le listener OpenMusic.

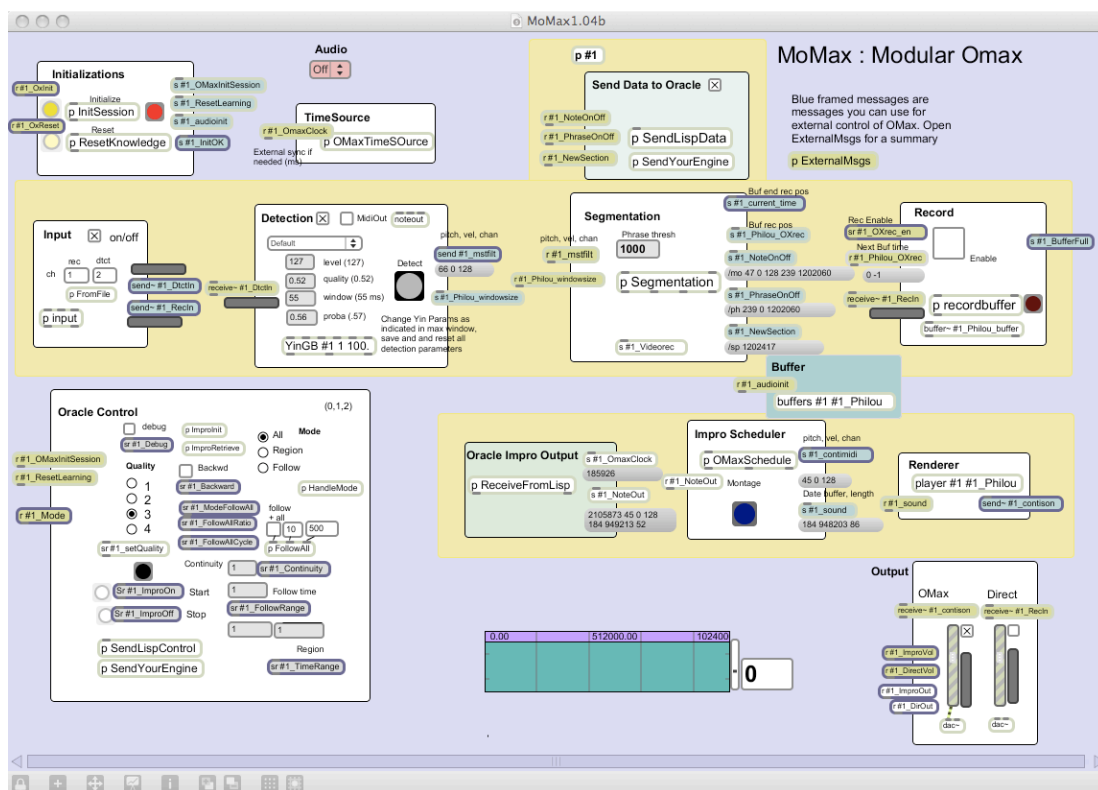
Ces améliorations ont été suscitées par un certain nombre de collaborations et d'expériences avec des musiciens : Bernard Lubat (acteur majeur de la scène des musiques improvisées), Dennis Thurmond (titulaire de la chaire d'improvisation dans l'école de musique réputée de l'University of Southern California), Mike Garson (improvisateur free lance américain, élève de Bill Evans et Herbie Hancock), François Nicolas, Philippe Leclerc, Georges Bloch. Une expérience avec Michel Portal n'a pas donné de grands résultats, ce dernier se disant, dans ses propres termes « terrorisé » d'affronter son double. Mais d'autres expériences ont été menées depuis avec des musiciens de premier plan : Fabrizio Cassol (Aka Moon, groupe réputé d'improvisation en Belgique), Alexandros Markeas (piano, dirige la classe d'improvisa-

tion du CNSM), Cécile Daroux (flûte), Vincent Lê-Quang (saxophone, CNSM), Laurent Mariusse (percussion).

Omax a été présenté lors d'ateliers-concerts à USC, à UCSD, à Harvard (Radcliffe Institute), à l'université Keio de Tokyo, au Forum Ircam, à NIME'06, à ICMC '07, au festival d'Uzeste, dans des interactions avec des professeurs et des élèves instrumentistes. Il a commencé à disséminer, d'autres groupes en proposant des implémentations alternatives à partir des algorithmes mis au point à l'Ircam. Un prototype réalisé par Shlomo Dubnov à UCSD exploite ainsi des descripteurs à base de MFCC (coefficients cepstraux) comme alternative aux descripteurs à base de pitch-tracking de la version Ircam. Cette version appelée OMax Spectral est maintenant intégrée dans la version courante [Bloch, Dubnov, Assayag 2008]. Alexandre François et Elaine Chew (USC) ont écrit une version basée sur leur système SAI (Software Architecture for ImmersiPresence), nommée MIMI (Multi-modal Interaction for Musical Improvisation) qui ajoute un feed-back visuel permettant au musicien d'anticiper les réactions de la machine (François, Chew, and Thurmond 2007). Omax et sa petite sœur MiMi ont été présentés conjointement lors d'ateliers à MCM'07 à Berlin, lors de l'édition inaugurale de la conférence Mathematical Computation in Music, issue (avec le Journal of mathematics and Music et la Society for Mathematics and Computation in Music) de la dynamique internationale initiée dans l'équipe Représentations Musicale autour des relations math-musique.



MoMax : OMax modulaire, interface



MoMax : OMax modulaire, patch révélant l'architecture d'OMax

15.4.3. OMax et le clonage

Le projet Omax est un cas particulièrement emblématique du problème de clonage : il s'agit en effet d'un système d'improvisation assistée par ordinateur, qui se propose de recréer le style de l'interprète, et de l'étendre à des développements ou des proliférations inattendues mais cohérents. Le fait de pouvoir confier une partie de la tâche d'improvisation à une machine a un énorme potentiel, puisqu'il touche directement à l'improvisation composée, ou plus précisément, au rapport entre l'improvisation et l'écriture. En effet, La machine peut connaître au préalable une partition écrite : l'improvisation commencera à partir de ce matériau. De plus, elle répond en fonction d'une loi de similarité qui peut, elle aussi, évoluer avec le temps en fonction d'une écriture.

Les recherches récentes ont porté sur un retour théorique au cœur de l'algorithme central d'OMax, l'oracle des facteurs. Cet algorithme a été initialement choisi parce qu'il permet de construire de manière incrémentale (donc compatible avec un déroulement causal et temps-réel) une représentation séquentielle des événements musicaux captés incorporant la connaissance quasi complète des structures de motifs (patterns). En parallèle à cette construction en ligne, un processus de *navigation* dans le modèle permet d'engendrer des séquences attribuées au clone. Or cette navigation

est critique pour la qualité musicale de la génération. Un travail théorique original a permis de comprendre la structure de l'oracle mieux qu'elle n'était décrite dans la littérature et de proposer des heuristiques de navigation innovantes [Assayag, Bloch 2007]. En particulier, les structures de flèches reliant les motifs forment une forêt d'arbres disjoints, représentant des ordres partiels sur la taille des motifs, qui décrivent complètement le contenu. Les parcours dans ces arbres, optimisés grâce à la structure d'ordre partiel (qui permet d'éliminer des sous-arbres entiers de la recherche) et à une distribution de probabilité favorisant les points de montage justifiés par des grands contextes communs, permettent une amélioration très notable de la qualité. Les arbres portent principalement sur les hauteurs et les textures.

Un second contrôle de qualité est effectué en vérifiant la cohérence rythmique, par un calcul de densité temporelle sur les contextes communs aux points de montage. Une des améliorations vérifiables de cette nouvelle stratégie de navigation est la stabilité formelle : si la séquence d'entrée est segmentable en grandes parties très auto-similaires individuellement et contrastées entre elles, le clone a tendance à se stabiliser tout seul dans une partie pour un certain temps, avant de s' « échapper » vers une autre partie dans laquelle il se stabilisera, etc.

Les musiciens ayant expérimenté avec les versions précédentes du système ont apprécié cette stabilité qui leur donne une sécurité formelle, et leur permet aussi de contrôler mieux le comportement du système. D'autre part, il a été constaté qu'un grand nombre des recombinaisons jugées autrefois « hasardeuses » par les experts étaient dues à des incohérences rythmiques.

Les parties Max et Lisp de OMax ont été récemment revues pour intégrer ces nouvelles stratégies et OMax a pu être distribué au Forum des utilisateurs ircam.

Parallèlement à ces améliorations, une réflexion a été engagée sur la partie « écriture ». Dans cette perspective, en plus de musiciens dont le jeu est capté et du développement d'un clone informatique (appelé aussi portrait du musicien), le système comprend un « texte » (concept proposé par Georges Bloch). Ce terme recouvre une stratégie formelle fixée à l'avance (concept plus général que celui de composition, mais comprenant notamment cette dernière). Le texte agit comme contrainte : soit qu'il constitue une partition fixée, soit qu'il constitue un ensemble de règles ou de directives. Le clone ne peut alors plus se développer librement comme dans la version « improvisation » mais doit tenir compte de cette contrainte pour que certaines propriétés définies au départ restent préservées (cohérence rythmique, harmonique, texturale, etc.). Le texte, plus généralement qu'une partition fixe, est donc un ensemble de contraintes, éventuellement modélisées par de la programmation logique (voir projet REACT). On peut alors imaginer que le développement du clone, s'il doit d'un côté respecter les contraintes, puisse d'un autre côté *remettre en question* ces dernières, donc changer le texte. Cette option audacieuse serait d'un certain point de vue un renouveau inédit de la notion de forme ouverte.

15.4.4. Enjeux cognitifs et anthropologiques

Ce projet évoque plusieurs enjeux cognitifs :

- la structure résultant de l'apprentissage est véritablement une structure de mémoire à long terme, et particulièrement de mémoire musicale. La stratégie de recall recombinaut à partir de cette mémoire (à l'origine du phénomène de réinjection stylistique) telle qu'elle est appliquée dans OMax se retrouve par exemple dans les phénomènes de glossolalies des rites initiatiques.
- les informations ne sont pas stockées de manière littérales dans cette mémoire : seules sont stockés les circuits qui permettent de reconstituer l'information de manière créative. Cela correspond à la vision moderne de la mémoire comme processus continu de recatégorisation (dans notre cas, la *variation*)
- les différents niveaux d'analyse dans OMax correspondent à des niveaux de catégorisation (clustering) locale et globale. Les descripteurs du signal sont d'abord agrégés temporellement selon leur similarité (détection de stabilité et segmentation) et dans un deuxième temps (pendant l'apprentissage) les unités élaborées au niveau précédent sont catégorisées pour former un vocabulaire (ou un lexique). Cette catégorisation est implicite dans le fonctionnement de l'oracle qui ne stocke pas statiquement un vocabulaire de labels mais reconstitue dynamiquement, à chaque nouvelle entrée la cartographie des occurrences de cette dernière dans la séquence déjà acquise. Toute la structure de l'oracle est en fait optimisée pour que ce travail cartographique soit extrêmement optimisé. De nouveau, le fonctionnement est proches des structures neurales qui fonctionnent plus par réactivation de circuits que par stockage.
- une variante du clustering local/global a été testée par Arshia Cont à l'Ircam dans le cadre de sa thèse (Cont 2008). Les unités élémentaires (des fenêtres d'analyse FFT) sont placées dans un cadre de géométrie de l'information [Cont, Dubnov, Assayag 2008], c'est à dire dans un espace où les points sont des distributions de probabilités. Les similarités sont alors évaluées par des divergences de Bregman qui, dans ce cas se, réduisent à des divergences informationnelles de Kulback-Leibler (information mutuelle). Les unités qui se groupent dans une même boule n-dimensionnelle (boule de Bregman) sont dans le même mouvement reconnues aux deux niveaux (comme un segment du flux analysé, comme un label du vocabulaire déjà rencontré). Cette technique est très prometteuse mais pas encore disponible en temps-réel.

16. Conclusion et perspectives

Au début de ce mémoire nous avons posé les deux termes composer et interagir comme constitutifs de l'informatique musicale. Plus précisément, il s'agissait de fournir des outils compositionnels au sens mathématique pour favoriser l'exploration expérimentale d'espaces créatifs, et des outils pour interagir avec ces espaces. Le facteur compositionnel ouvre à la maîtrise de la complexité. L'interaction ouvre aux interfaces de navigation dans les structures complexes et à la mise en situation de ces structures dans une performance vivante.

Bien entendu, c'est une cohérence rendue après-coup. Le parcours dans ce document montre une variété hétérogène de thématiques de recherches que je n'ai pas cherché à déguiser en homogénéité factice. Ces expériences sont comparables aux *rhumbs* de Paul Valéry. « Les remarques et les jugements qui composent ce livre me furent autant d'écarts d'une certaine direction privilégiée de mon esprit: d'où Rhumbs » (Valéry, Œuvres, t. 2, Tel quel II, Rhumbs, Paris, Gallimard, 1960 [1926], p. 597). Le mot est un ancien terme de marine qui désigne un petit angle, celui que négocie par exemple un esquif contre le vent dans une série de virages opposés qui lui permettent de maintenir la direction globale. Toutes ces thématiques sont donc des rhumbs, dont je puis mesurer après-coup qu'ils festonnent une « certaine direction privilégiée » et sous-tendent l'intuition d'une unité à viser de l'informatique musicale, et même plus généralement de l'informatique pour la création.

Le parcours nous a mené de la formalisation de la musique et de sa notation à l'intégration progressive de ses constituants sonores et symboliques, de son être comme *structure* et de son existence comme *processus*.

Ce parcours a été long parce que certaines évolutions scientifiques et technologiques devaient s'accomplir avant que ne puisse se concrétiser de telles intégrations. Par exemple il a fallu que certains problèmes du temps réel soient réglés (vitesse de calcul, environnements portables à hautes performances) pour évacuer l'obsession du temps réel et redécouvrir une vérité paradoxale : l'être humain (qui reste le modèle des exploits que l'on cherche à accomplir avec une machine) est finalement assez lent. Les expériences en psychologie cognitive montrent que les temps de réactions basiques à certains stimuli se comptent en centaines de millisecondes. Les latences visées actuellement par les systèmes de traitement du signal en temps réel se comptent en millisecondes. Que faire de tout ce temps disponible ? Tout à coup, un ensemble de sous-domaines de l'informatique musicale, rangés, quelquefois avec condescendance, dans la catégorie « temps différé » donc incompatible avec la performance musicale, et qui ont progressé de leur côté dans l'efficacité algorithmique et l'expressivité symbolique, peuvent venir se nicher dans cet interstice. En quelques centaines de millisecondes, peuvent s'exécuter des algorithmes d'informatique symbolique, des processus d'optimisation complexes, des schémas de programmation

logique avec contraintes, qui ne sont pas temps-réel stricto sensu mais peuvent être perçus comme tels. Un improvisateur qui prend une demi-seconde pour réaliser que le contexte musical vient de changer et transformer la note qu'il était en train de jouer de manière qu'elle s'adapte au nouveau contexte est perçu comme producteur d'une réaction instantanée.

Nous parlons là de réaction au changement. Dans le cas de la synchronisation en régime stable (synchronisation des musiciens au tempo d'un percussionniste par exemple) le phénomène d'instantanéité résulte de *l'anticipation musicale*. Enfin, la stratégie de développement, projection vers le futur, se rapproche, elle, de la composition en ce qu'elle agence une structure virtuelle (mise à jour en permanence) qui rend compte de la situation à la fois passée et future et fournit un cadre à l'anticipation.

Nous parlons donc d'anticipation, de réaction au changement, et de stratégie compositionnelle, ces trois activités étant conflictuelles. Sur le plan informatique, il s'agit de régler de manière concurrente le ballet de ces trois activités qui interagissent sur des échelles temporelles spécifiques. OMax est déjà une prémisse de ce principe, dans la mesure où son composant Max règle la réactivité immédiate et la « perception artificielle » du musicien, alors que son composant OpenMusic règle concurrentement la modélisation de haut niveau, la reconnaissance de structures et les stratégies de production.

Nous voulons cependant aller plus loin.

Nous ne sommes qu'au début des possibilités qu'offrent la concurrence (et de son substrat technique le parallélisme) et les modèles cognitifs (cf. en particulier la vogue actuelle de la modélisation computationnelle dirigée par la cognition, *cognitive driven*). Il y a cependant une difficulté. Si on voit bien comment une meilleure modélisation cognitive pourrait améliorer un système comme OMax (par exemple par une intégration de la mémoire à court terme qui lui donne une meilleure réactivité au changement), il n'est pas certain que tout système d'aide à la création soit in fine une imitation du système cognitif. En effet nous visons des systèmes d'aide à la découverte, des systèmes d'exploration. Les surprises issues du fonctionnement algorithmique participent elles aussi de la création.

Nous envisageons l'évolution de tels environnements vers des systèmes complexes avec lesquels l'utilisateur interagira en donnant lieu à des phénomènes d'émergence structurelle. L'idée vient en particulier du travail sur l'orchestration (thèse de Grégoire Carpentier) avec lequel nous sommes déjà dans une situation d'interaction avec le modèle d'un gigantesque « orchestre » où des processus évolutionnistes font émerger les solutions qui peuplent un front de Pareto. Nous nous sommes alors dit que ce principe pourrait être étendu au-delà de l'orchestration (qui n'est qu'un problème local lié à la composition) à la composition en général. Ainsi une composition

serait vue comme la « réponse » d'un système dynamique (l'orchestre) aux « stimuli » du compositeur qui essaie de susciter des structures et des formes musicales obéissant à certaines contraintes. La valeur ajoutée d'une telle approche prospective tiendrait à ce que dans le même mouvement seraient résolus les problèmes formels et les problèmes sonores (l'évolution de l'orchestre ne pouvant aller que dans des directions qui respectent certaines cohérences timbrales). On voit là que c'est une réponse au réductionnisme dont nous sommes bien conscients maintenant qu'il constitue le verrou principal pour la question des échelles multiples de temps et de structures.

Il faut cependant être prudent, car le risque est alors de perdre la propriété de compositionnalité des espaces engendrés par l'expérimentation (caractéristique dont nous avons à plusieurs reprises mentionné qu'elle nous semblait indispensable).

Nous n'avons pas encore la réponse, mais une piste, peut-être : le travail d'Andrée Ehresmann, mathématicienne qui travaille depuis plus de vingt ans sur les « Systèmes évolutifs à mémoire » (Ehresmann & Vanbremeersch 2007), dresse un pont entre les sciences de la complexité, en pleine ébullition, et l'algèbre la plus abstraite. En revisitant les catégories classiques (auto-organisation, émergence) à l'aide de concepts algébriques de la théorie des catégories (limites, co-limites, foncteurs), elle propose un formalisme compositionnel permettant de rendre compte des conditions d'évolution de structures neurales. C'est dans l'idéal le meilleur des deux mondes, mais un chemin similaire vers la musique n'est pas encore tracé.

17. Sélection de pièces utilisant OpenMusic

Michel Amoric, Tombeaux de Marin Mersenne pour théorbe et synthétiseur, 2005

Alain Bancquart, Amour grand terrible champ critique, pour percussion et électronique, 2004

Georges Bloch Empreinte sonore de la fondation Bayeler, pour ensemble, 2004

Elvio Cippolone, Concerto pour basson et électronique, 2006

Shlomo Dubnov Memex et Composer Duets pour violons, 2006

Brian Ferneyhough Stellæ for failed times, pour chœur et électronique, 2001

Joshua Fineberg, Lolita pour ensemble, chorégraphie et vidéo, 2006-2008, The texture of time pour flûte et électronique, 2006

Karim Haddad, Livre Premier de Motets, pour deux à cinq voix, 2007, First Attempted Escape from Silence: Tunnels, acte d'opéra sur un livret de Jonathan Safran Foer, 2005

Jean-Luc Hervé, Encore pour orchestre et pianos mécaniques, 2000

Philippe Hurel, Hors-jeu pour percussions et électronique, 2006

Michael Jarrel, Droben Schmettert ein greller stein pour contrebasse, ensemble et électronique, 2001

Johannes Kretz, Second Horizon pour Piano and Orchestre, 2002

Mauro Lanza, Aschenblume pour neuf instruments (2001-2002), Erba nera che cresci segno nero tu vivi pour soprano et sons de synthèse (1999)

Philippe Leroux VOI(REX), voix, ensemble et électronique, 2002

Fabien Lévy, Hérédo-Ribotes pour orchestre, 2001, Coïncidences pour orchestre, 1999

PerMagnus Lindborg, TreeTorika, pour orchestre de chambre , 2006

Grégoire Lorieux Langage de l'ombre pour voix, hautbois, basson, clarinette, clarinette basse, contrebasse et électronique, 2006

Claudy Malherbe, Locus pour six voix, 1997, Blanc-noir, grisé pour ensemble, 2002, La Cantatrice, opéra radiophonique, 2008

Mikhail Malt, Six Fractal Contemplations, 2004

Philippe Manoury, K... opéra en douze scènes, pour voix solistes, orchestre et électronique en temps réel (2000-2001)

Simon Mawhinney, Starbog pour flute, clarinette, cymbalum, percussion, piano, violon et violoncelle, 2006

Christopher Melen, Six Metal Fugue, pour quatre cuivres, cloches tubulaires et steelpans, 2006

Tristan Murail, Terre d'ombre pour grand orchestre et sons électroniques (2003-2004), Contes cruels pour 2 guitares électriques et orchestre (2007)

Paul Nauert, A Collection of Caprices pour piano, 2003

Ketty Nez, Sculpted Implosions, pour cor et électronique, 1999

Hèctor Parra, Strette, monodrame pour soprano, électronique, vidéo en temps réel et lumières (2000-2003),

Luís Antunes Pena, Klangspiegel pour trompette en quart de ton, tam tam et bande

Kaija Saariaho's, L'amour de loin, Opéra, 2000

Orjan Sandred, Kalejdoskop pour clarinette, alto et piano, 1999

Asbjørn Schaathun Double portrait, concerto pour violon avec électronique, 2006

António de Sousa Dias, Dói-me o luar pour ensemble, 2001, Ressonâncias - Memórias, pour ensemble et bande, 2003

Hans Tutschku, Distance liquide, musique électroacoustique huit canaux, 2007

Tolga Tüzün, Metathesis, pour contrebasse et électronique, 2006

18. Publications personnelles / sous ma direction

18.1. Journaux, chapitres de livres et livres

Agon, C., Andreatta, M., Assayag, G., Schaub, S., « Formal Aspects of Iannis Xenakis' "Symbolic Music": A Computer-Aided Exploration of Compositional Processes », *Journal of New Music Research*, Juin 2004, vol. 33, n° 2, pp. 145-159

Agon, C., Assayag, G., (2003a) « Outils informatiques pour la composition musicale assistée par ordinateur : un état de l'art. », in Baboni Schilingi J. (ed.), *Book of articles and essays of PRISMA group*, Euresis Edizioni, Milano, 2003

Agon, C., Assayag, G., « Object-Oriented Programming in OpenMusic », in Mazzola G. (ed.), *The Topos of Music*, Birkhäuser Verlag, 2002

Assayag, G., « De la calculabilité à l'implémentation musicale », in Assayag, G., Nicolas, F. et Mazzola G. (eds.) *Penser la musique avec les mathématiques*, Collection Musique et Sciences, Coédition Ircam - Centre Pompidou / Delatour France, 2006.

Assayag, G. (Ed.), *Journal of New Music Research, Special Issue on Selected Papers from the Sound and Music Computing Conference '04*, Volume 34, Number 4, Taylor & Francis, December 2005

Assayag, G., « Computer », *Cahiers de médiologie*, Novembre 2004, n° 18

Assayag, G., *Accordes y desacuerdos, transformaciòn afin*, in *Fotografiando las matemáticas*, in Guerin S., Guerin P., Puig, M. (eds.), Carroggio, Barcelona, 2000.

Assayag, G., « Du calcul secret au calcul visuel », in François Delalande et Hugues Vinet (eds.), *Interfaces homme-machine et création musicale*, Hermès, Paris, 1999

Assayag, G., *A matematica, o numero e o computador. Coloquio / Ciencias*, Revista de Cultura Científica, Lisboa, 1999, n° 24

Assayag, G., «Composition Assistée par Ordinateur : nouveaux chemins de création», Gérard Assayag. :éc/artS : (1) E. Sadin éditeur, Roanne 1998.

Assayag, G., «Composition assistée par ordinateur». Vers la partition potentielle. Les Cahiers de l'Ircam. Ircam-Centre Georges Pompidou ed. 1993.

Assayag, G. «Informatique et création contemporaine», Les Lettres Nouvelles, Paris Février 1991.

Assayag, G., Agon, C., « OpenMusic : un langage de programmation visuelle pour la composition », in Chemillier, M., Pachet F. (eds.), *Encyclopédie pour l'ingénieur informaticien*, Hermès, 2000

Assayag, G., Bloch, G., Quantification et création musicale, in M. Grabocz (ed), Les modèles dans l'art, musique, peinture, cinéma. Presses Universitaires de Strasbourg, 1997.

Assayag, G., Bresson, J., « OpenMusic : de la composition à l'enseignement », L'Inouï, 2006, n° 2

Assayag, G., Cafagna, V., Chemillier M. (eds.), Soft Computing - a Fusion of Foundations, Methodologies and Applications, Special Issue on Formal Systems and Music, 8(9), 2004

Assayag A., Cholleton J.P., Musique, nombres et ordinateurs, La Recherche, Paris, Juillet / Août 1995 n° 278

Assayag, G., Cholleton, J.P., «L'appareil musical». Résonance, Ircam-Centre Georges Pompidou, Octobre 1994.

Assayag, G., Dubnov, S., « Using Factor Oracles for machine Improvisation », Soft Computing - a Fusion of Foundations, Methodologies and Applications, Septembre 2004, vol. 8, n° 9, pp. 604-610

Assayag, G., Malherbe, C., Cinq questions sur l'audibilité en musique, Conséquence Paris, 1986, n° 7-8

Assayag, G., Nicolas, F., Mazzola, G. (eds.), «Penser la musique avec les mathématiques», Actes du séminaire mathématiques / musique / philosophie, Collection musique et sciences, Coédition Ircam - Centre Pompidou / Delatour France, 2006.

Assayag, G., Rodrigues J.F., & Feichtinger H., (eds.), Mathematic and Music, Springer-Verlag, Berlin, 2002

Assayag, G., Rueda, C., «Représentations Musicales». Intermedia, Barcelone 1993.

Assayag, G., Rueda, C., Laurson, M., Agon, C., Delerue, O., « Computer Assisted Composition at Ircam : patchWork & OpenMusic », Computer Music Journal, Décembre 1999, vol. 23, n° 5, pp. 59-72

Bresson, J., Agon, C., Assayag, G. (eds.), « The OM Composer's Book : volume 2 », The OM Composer's Book series, Collection musique et science, co-éditions Delatour France / Ircam-Centre Pompidou, 2008

Bresson, J., Agon, C., Assayag, G. (eds.), « The OM Composer's Book : volume 1 », The OM Composer's Book series, Collection musique et science, co-éditions Delatour France / Ircam-Centre Pompidou, 2006

Bresson, J., Guédy, F., Assayag, G., « Musique Lab Maquette : approche interactive des processus compositionnels pour la pédagogie musicale », STICEF - Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, 2006, n° 13

- Carpentier, G., Assayag, G., Saint-James, E. « Solving the Musical Orchestration Problem using Multiobjective Constrained Optimization with a Genetic Local Search Approach », *Journal of Heuristics*, accepted, to appear 2009.
- Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E., (2007a) « An Evolutionary Approach to Computer-Aided Orchestration », *Applications of Evolutionary Computing, Lecture Notes in Computer Science*, Springer, 2007, pp. 488-497
- Carpentier, G., Tardieu, D., Harvey J., Assayag, G., Saint James, E., «Constrained Multiobjective Optimization for Computer-Aided Orchestration: Theory and Applications», *Journal of New Music Research*, submitted dec. 2008
- Chemillier, M. & Assayag, G.. « OMax : présentation multimédia des recherches sur l'improvisation et l'ordinateur de l'Ircam et de la Cie Lubat », *Musimédiante*, « Musiques non écrites », mai 2008, n° 3
- Cont, A., Dubnov, S., Assayag, G. «On the Information Geometry of Audio Streams with Applications to Automatic Structure Discovery», *IEEE TRANS. ON AUDIO, SPEECH AND LANGUAGE PROC.*, DECEMBER 2008, submitted.
- Cont, A., Dubnov, S., Assayag, G., « Anticipatory Model of Musical Style Imitation using Collaborative and Competitive Reinforcement Learning », in Butz, M., Sigaud O., and Baldassarre G. (eds.), *Anticipatory Behavior in Adaptive Learning Systems*, *Lecture Notes in Computer Science*, Springer Verlag, Berlin, 2007, vol. 4520, pp. 285-306
- Daubresse, Eric, Assayag, Gerard. «Technology and Creation - The Creative Evolution», *Contemporary Music Review*, Routledge, London , 2000, Vol. 19, n° 2, pp. 61-80.
- Dubnov, S., Assayag, G., Lartillot, O., Bejerano, G., « Using Machine-Learning Methods for Musical Style Modeling », *IEEE Computer*, Octobre 2003, vol. 10, n° 38, pp. 73-80
- Dubnov, S., Assayag, G., « Universal Prediction Applied to Stylistic Music Generation », in Assayag, G., Feichtinger, H.G., Rodrigues, J.F. (eds.), *Mathematics and Music*, Springer-Verlag, Berlin, 2002, pp. 147-158
- Lartillot O., Assayag, G., "Modélisation et simulation du système complexe de perception de structures musicales", in Mondher Ayari (éd.), *De la théorie à l'art de l'improvisation : Analyse de performances et modélisation musicale*, Delatour France, 2005.
- Rueda, C., Alvarez, G., Quesada, L., Tamura, G., Valencia, F., Diaz Frias, J., Assayag, G., « Integrating Constraints and Concurrent Objects in Musical Applications : a calculus and its Visual Language », *Constraints Journal*, Kluwer Academic Publisher, 2001, vol. 6, n° 1, pp. 21-52

18.2. Conférences avec comité de programme

Agon, C., Assayag, G., (2003b) « OM :A Graphical extension of CLOS using the MOP », International Lisp Conference ILC, New York, 2003.

Agon, C., Assayag, G., « Programmation Visuelle et Editeurs Musicaux pour la Composition Assistée par Ordinateur. », Interfaces Homme Machine IHM'02, Poitiers, 2002

Agon, C., Assayag, G., Delerue, O., Rueda, C., « Objects, Time and Constraints in OpenMusic », ICMC: International Computer Music Conference, Ann Arbor Michigan, 1998

Agon, C., Assayag, G., Fineberg, J., Rueda, C., « KANT : a Critique of Pure Quantification », International Computer Music Conference, Aarhus, 1994, pp. 52-59

Agon, C., Bresson, J., Assayag, G., « OpenMusic: Design and Implementation Aspects of a Visual Programming Language », 1st European Lisp Symposium - ELS'08, Bordeaux, 2008

Agon, C., Haddad, K., Assayag, G., « Representation and Rendering of Rhythmic Structures », Intl. Conf. On Delivering of Music, WedelMusic '02, Darmstadt, 2002

Agon, C., Stroppa, M., Assayag, G., « High Level Musical Control of Sound Synthesis in OpenMusic », International Computer Music Conference, Berlin, 2000

Allombert, A., Assayag, G., Desainte-Catherine, M., « A system of Interactive Scores based on Petri Nets », SMC'07 - 4th Sound and Music Computing Conference, Lefkada, 2007, pp. 158-165

Allombert, A., Assayag, G., Desainte-Catherine, M., Rueda, C., « Concurrent Constraints Models For Interactive Scores », Sound and Music Computing SMC'06, Marseille, 2006

Allombert A., Desainte-Catherine M. et Assayag G., «De Boxes à Iscore : vers une écriture de l'interaction», Actes des 13èmes Journées d'Informatique Musicale (JIM08), Albi, France, Mars 2008.

Allombert, A., Desainte-Catherine, M., and Assayag, G. «Iscore: a system for writing interaction». In Proceedings of the 3rd international Conference on Digital interactive Media in Entertainment and Arts (Athens, Greece, September 10 - 12, 2008). DIMEA '08, vol. 349. ACM, New York, NY, 360-367

Allombert A., Desainte-Catherine M., Larralde J. and Assayag G., «A system of Interactive Scores based on qualitative and quantitative temporal constraints», Proc. of the 4th International Conference on Digital Arts (Artech 2008), Porto, Portugal, November 2008.

- Alvarez, G., Diaz, J.F., Quesada, L.O., Valencia, F., Rueda, C., Assayag, G. «PICO : a Calculus of Concurrent Constraint Objects for Musical Applications», ECAI 98, workshop « Constraint techniques for artistic applications », Brighton, 1998.
- Andreatta, M., Noll, T., Agon, C., Assayag, G., « The Geometrical Groove : rhythmic canons between Theory, Implementation and Musical Experiments », Journées d'Informatique Musicale JIM'2001, Bourges, 2001, pp. 93-98
- Aranda, J., Assayag, G., Olarte, C., Perez, J.A., Rueda, C., Toro M., Valencia, F. «An Overview of FORCES: An INRIA Project on Declarative Formalisms for Emergent Systems», ICLP09, 2009
- Assayag, G., « Computer Assisted Composition Today », 1st Symposium on Music and Computers, Corfu, 1998
- Assayag, G., « Visual Programming in Music », International Computer Music Conference ICMC'95, Banff, 1995, pp. 73-76
- Assayag G., Agon, C., «OpenMusic Architecture». International Computer Music Conference ICMC'96, Hong Kong 1996.
- Assayag, G., Agon, C., Fineberg, J., Hanappe, P., « An Object Oriented Visual Environment for Musical Composition », ICMC: International Computer Music Conference, Thessaloniki, 1997, pp. 364-367
- Assayag, G., Agon, C., Hanappe, P., «Problèmes de notation dans la composition assistée par ordinateur». Acte des Rencontres Musicales Pluridisciplinaires, Musique et Notation, GRAME, Lyon 1997
- Assayag G., Hanappe, P. «Problèmes de quantification et de transcription en composition assistée par ordinateur». Actes des rencontres Musicales Pluridisciplinaires, Musique & Mathématiques, H. Genevois et Y. Orlarey éditeurs, GRAME, Lyon 1996.
- Assayag, G., Amiot E., Malherbe C., Riotte A., «Duration Structures Generation and Recognition in Musical Writing», International Computer Music Conference ICMC'86, La Haye 1986
- Assayag, G., Bejerano, G., Dubnov, S., Lartillot, O., « Automatic modeling of musical style », Journées d'Informatique Musicale JIM'2001, Bourges, 2001, pp. 113-119
- Assayag, G., Bloch, G., « Navigating the Oracle: a Heuristic Approach », International Computer Music Conference '07, Copenhagen, 2007, pp. 405-412
- Assayag, G., Bloch, G., Chemillier, M., « Improvisation et réinjection stylistiques », Le feed-back dans la création musicale contemporaine - Rencontres musicales pluridisciplinaires, Lyon, 2006
- Assayag, G., Bloch, G., Chemillier, M., « OMax-Ofon », Sound and Music Computing SMC'06, Marseille, 2006

- Assayag, G., Bloch, G., Chemillier, M., Cont, A., Dubnov, S., « Omax Brothers : a Dynamic Topology of Agents for Improvization Learning », Workshop on Audio and Music Computing for Multimedia, ACM Multimedia '06, Santa Barbara, 2006
- Assayag, G., Bloch, G., Jullien, J.P., Warusfel, O., «Spatializer : from Room Acoustics to Virtual Acoustics», International Computer Music Conference ICMC'92, San Jose, 1992.
- Assayag, G., Castellengo, M., Malherbe C., (1985a) «Functional Integration of Complex Instrumental Sounds in Music Writing», International Computer Music Conference ICMC'85, Vancouver 1985
- Assayag, G., Dubnov, S., « Improvisation Planning and Jam Session Design using concepts of Sequence Variation and Flow Experience », Sound and Music Computing 2005, Salerno, 2005
- Assayag, G., Dubnov, S., Delerue, O., « Guessing the Composer's Mind : applying Universal Prediction to Musical Style », ICMC: International Computer Music Conference, Beijing, 1999
- Assayag, G., Rueda, C., « The Music Representation Project at IRCAM », ICMC: International Computer Music Conference, Tokyo, 1993
- Assayag, G., Timis, D., «Computer Printing, Storage, and Transfer of Musical Scores», International Computer Music Conference ICMC'87, Urbana Illinois, 1987
- Assayag, G., Timis, D., «A ToolBox for Music Notation», International Computer Music Conference ICMC'86, Den Haag, 1986.
- Bloch, G., Dubnov, S., Assayag, G., « Introducing Video Features and Spectral Descriptors in the OMax Improvisation System », International Computer Music Conference ICMC'08, Belfast, 2008
- Bresson, J., Agon, C., Assayag, G., « OpenMusic 5: A Cross-Platform Release of the Computer-Assisted Composition Environment », 10th Brazilian Symposium on Computer Music, Belo Horizonte, 2005
- Carpentier, G., Bresson, J. «Interfaces for Computer-Aided Orchestration: Linking Heterogeneous Spaces with User Interaction», Computer Music Journal, submitted 2008
- Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E., (2007b) « An Evolutionary Approach to Computer-Aided Orchestration », EvoMUSART, vol. LNCS 4448, Valence, 2007, pp. 488-497
- Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E., « Imitative And Generative Orchestration Using Pre-Analysed Sounds Databases », SMC'06, Marseille, 2006, pp. 115-122

- Cont, A., Dubnov, S., Assayag, G., « GUIDAGE: A Fast Audio Query Guided Assemblage », International Computer Music Conference, Copenhagen, 2007
- Cont, A., Dubnov S., Assayag, G., "A Framework for Anticipatory Machine Improvisation and Style Imitation", Adaptive Behavior in Anticipatory Learning Systems in association with the conference Simulation of Adaptive Behavior (SAB 2006) Rome, Italy, 2006.
- Dubnov, S., Assayag, G., El-Yaniv, R. «Universal Classification Applied to Musical Sequences», Proceedings of the ICMC'98, Ann Arbor, Michigan 1998, pp. 332-340.
- Dubnov, S., Cont, A., Assayag, G., « Audio Oracle: A New Algorithm for Fast Learning of Audio Structures », International Computer Music Conference, Copenhagen, 2007
- Guédy, F., Bresson, J., Assayag, G., « Musique Lab 2 - Un environnement d'aide à la pédagogie musicale », Journées d'Informatique Musicale - JIM'07, 2007
- Hanappe, P., Assayag, G., « Composition on Top of Real-Time» Actes des JIM98, Journées d'Informatique Musicale, Lalonde-les-maures 1998.
- Hanappe, P., Assayag, G., «A proposition of data structure and a framework for computer aided composition and sound synthesis». Journées d'Informatique Musicale JIM'97, Lyon 1997.
- Lartillot, O. , Ayari, M. , Assayag, G. , McAdams S., « Segmentation et reconnaissance de patterns dans la musique », Actes des journées (SCO/0039) : Action Concertée Incitative (ACI Cnrs) « Systèmes complexes en Sciences sociales», Paris, 2004
- Lartillot, O., Dubnov, S., Assayag, G., Bejerano, G., « Automatic Modeling of Musical Style », International Computer Music Conference, La Havane, 2001, pp. 447-454
- Rueda, C., Assayag, G., Dubnov, S., « A Concurrent Constraints Factor Oracle Model for Music Improvisation », XXXII Conferencia Latinoamericana de Informática CLEI 2006, Santiago, 2006
- Rueda, C., Assayag, G., Laurson, M., Bloch, G., « Integrating Constraint Programming in Visual Musical Composition Languages », ECAI 98, workshop « Constraint techniques for artistic applications », Brighton, 1998.
- Truchet, C., Agon, C., Assayag, G., « Recherche adaptative et contraintes musicales », 10ème Journées francophones de programmation logique et programmation par contraintes JFPLC'01, Paris 2001
- Truchet, C., Agon, C., Assayag, G., Codognet, P. " CAO et contraintes " Journées d'Informatique Musicale JIM'2001, Bourges, 2001.
- Truchet, C., Assayag, G., Codognet, P., (2003a)« Musical Application of Adaptive Search, a Tabu Search Method for Solving CSPs », INFORMS, Cluster : OR in the Arts, Atlanta, 2003

Truchet, C., Assayag, G., Codognet, P., (2003b) « OMClouds, a Heuristic Solver for Musical Constraints », Metaheuristics International Conference MIC03, Kyoto, 2003

Truchet, C., Assayag, G., Codognet, P., (2003c) « OMClouds, petits nuages de contraintes dans OpenMusic », Journées d'Informatique Musicale JIM'03, France, 2003

Truchet, C., Assayag, G., Codognet, P., « Visual and Adaptive Constraint Programming in Music », International Computer Music Conference, La Havane, 2001

Truchet, C. , Diaz, D. , Codognet P. , « Résolution de Contraintes par Recherche Adaptative et Application à des CSP's musicaux », ROADEF03, Congrès de la Société Française de Recherche Opérationnelle, France, 2003

18.3. Rapports

Assayag, G., Mémoire pour l'obtention du DEA Langages, Algorithmes, Programmation, Laboratoire d'Informatique théorique et programmation, UPMC Paris 6, 1985

Assayag, G. Rapport scientifique annuel de l'équipe Représentation Musicale, 1992 - 2008

Assayag, G., «Le musicologue», rapport pour un système d'apprentissage de la musique sur micro-ordinateur, Act Informatique, 1987

Assayag, G., Castellengo, M., Malherbe, C., (1985b) «Elaboration de champs harmoniques dans la composition assistée par ordinateur», Rapport IRCAM n° 38. IRCAM, Paris décembre 1985

Assayag, G., Rodet, X., « Rapport sur les travaux du groupe Outils pour l'analyse musicale », Ircam, 2002

Assayag, G., Farre, J., « L'interprète (semi) graphique MilPil », Laboratoire Informatique Théorique et Programmation (LITP), UPMC, Rapport N° 84-32, Juin 1984.

18.4. Travaux d'étudiants encadrés

Agon, C., «Langages de programmation pour la composition musicale». Habilitation à Diriger des Recherches, Université Pierre et Marie Curie (Paris VI), 2004, co-direction avec Jean-Pierre Briot.

Agon, C., «OpenMusic : Un langage visuel pour la composition musicale assistée par ordinateur». Thèse de doctorat de l'Université Pierre et Marie Curie, Paris 6, 1998. Co-direction avec Jean-François Perrot.

Agon, C., «Kant : Une critique de la quantification pure». DEA d'informatique de l'université Paris XI. 1994.

Allombert, A., «Partitions interactives». Thèse de Doctorat de l'université Bordeaux II, soutenance 2009. Co-direction avec Myriam Desainte-Catherine.

Andreatta, M., «Méthodes algébriques dans la musique et la musicologie du XXème siècle : aspects théoriques, analytiques et compositionnels», Thèse de l'Ecole des Hautes Etudes en Sciences Sociales. Paris, 2003. Sous la direction de Hugues Dufourt et Marc Chemillier.

Andreatta, M., «La théorie mathématique de la musique de Guerino Mazzola et les canons rythmiques», Mémoire de DEA, Musique et Musicologie du Xxème siècle, Paris IV, EHESS, 1999, sous la direction de Hugues Dufourt et Marc Chemillier

Avitabile, F., «Cribles en Informatique Musicale». Magistère d'Informatique de l'Ecole Normale Supérieure de Lyon et de l'Université Claude Bernard (Lyon I), 1993

Brown, A. R., «Music composition and the computer : an examination of the work practices of five experienced composers», Ph. D., School of Music, University of Queensland, Brisbane, 2003, co-direction comme foreign supervisor.

Carpentier, G., «Approche computationnelle de l'orchestration musicale : optimisation multicritère sous contraintes de combinaisons instrumentales dans de grandes banques de sons ». Thèse de Doctorat de l'université Pierre et Marie Curie Paris VI, 2008. Co-direction avec Emmanuel Saint-James.

Delerue, O., «Spatialisation du son et programmation par contraintes : le système MusicSpace», Thèse de l'UPMC, 2004. En co-direction avec Jean-François Perrot et François Pachet.

Delerue, O., «Etude et réalisation d'opérateurs musicaux pour un environnement de composition assistée par ordinateur». DEA ATIAM, Ircam - Université Paris VI, 1997.

Durand, N., «Apprentissage du style musical et interaction sur deux échelles temporelles» DEA ATIAM, Université Pierre et Marie Curie, Paris VI, 2003

Eav, D., «OpenMusic et la notion de geste», mémoire d'ingénieur ICAM Toulouse, 2002. Direction Carlos Agon.

Guenego, J. L., «Une application de résolution de contraintes en harmonie tonale». DEA ATIAM, Ircam - Université Paris VI, 1998

Hanappe, P., «Design and implementation of an integrated environment for music composition and synthesis». Thèse de doctorat de l'Université Pierre et Marie Curie, Paris 6, 1999. Co-direction avec Emmanuel Saint-James.

Hanappe, P., «Intégration des représentations temps/fréquence et des représentations musicales symboliques». DEA ATIAM, Ircam - Université Paris VI, 1995.

- Lartillot-Nakamura O., «Fondements d'un système d'analyse musicale computationnelle suivant une modélisation cognitive de l'écoute», Thèse de doctorat de l'Université Pierre et Marie Curie, Paris 6, 2004. Co-direction avec Emmanuel Saint-James.
- Lartillot, O., «Modélisation du style musical par apprentissage statistique : une application de la théorie de l'information à la musique», DEA ATIAM, Ircam, 2000
- Laurier, C., «Attributs multiples dans un improvisateur automatique». DEA ATIAM, Université Pierre et Marie Curie, PARIS VI, 2004
- Maniatakos, F., «Nouveaux paradigmes d'interaction musicien-ordinateur», Thèse de Doctorat de l'université Pierre et Marie Curie Paris VI, soutenance 2010. Co-direction avec Carlos Agon et Frederic Bevilacqua.
- Mathieu, B., «Outils informatiques d'analyse musicale». Benoit Mathieu. DEA MIASH, ENST Bretagne, 2002
- Meudic B., «Détermination automatique de la pulsation, de la métrique et des motifs musicaux dans des interprétations à tempo variable d'oeuvres polyphoniques». Thèse de doctorat de l'Université Pierre et Marie Curie, Paris 6, 2004. Co-direction avec Emmanuel Saint-James.
- Meudic, B., «Modélisation de structures rythmiques», DEA ATIAM, Ircam, 2000.
- Meudic, B., «Organisation rythmique dans un environnement d'aide à la composition», Mémoire d'ingénieur de l'IE, 1999.
- Nouno, G., «Suivi de Tempo Appliqué aux Musiques Improvisées : à la recherche du temps perdu ...». Thèse de Doctorat de l'université Pierre et Marie Curie Paris VI, 2008. Co-direction avec Philippe Codognet.
- Poirson, E., «Simulations d'improvisations à l'aide d'un automate de facteurs et validation expérimentale», DEA ATIAM, Paris 6, Juillet 2002, en co-direction avec Emmanuel Bigand
- Rokita, L., «Modèle informatique d'une pièce pour clarinette d'Igor Stravinsky». DEA ATIAM, Ircam - Université Paris VI, 1994
- Toro Bermudez, M., « Exploring the possibilities and limitations of Concurrent Programming for Multimedia Interaction and exploring Visual Programming for Musical Constraint Satisfaction Problems », Université Pontificale Javeriana de Cali, Ircam, mémoire de master, 2008, co-direction avec Camilo Rueda.
- Truchet, C., «Contraintes, recherche locale et composition assistée par ordinateur», Thèse de Doctorat de l'université Paris VII, 2004. Co-direction avec Philippe Codognet.

Valèze, B., «Intégration d'un système de contraintes temporelles dans le logiciel de composition musicale OpenMusic. Master II en Informatique, Université Bordeaux I - Scrim / Ircam, 2006, en co-direction avec Myriam Desainte-Catherine

Verière, V., «Kant : Quantification Rythmique Structurée», Veronique Verière, DESS d'Intelligence Artificielle PARIS VI, 1993.

18.5. Logiciels

OMax, 2003 à aujourd'hui, est un environnement permettant la construction en temps réel d'un modèle stylistique à partir du jeu d'un ou de plusieurs musiciens. Il peut être utilisé à des fins d'analyse (caractéristiques du style, étude de la mémoire musicale) et à des fins créatives comme un partenaire synthétique pour l'improvisation ou la composition. OMax repose sur OpenMusic et Max et a suscité de nombreuses publications. Avec Georges Bloch et Marc Chemillier.

OpenMusic, 1993 à aujourd'hui, a été conçu à l'Ircam par Carlos Agon et Gérard Assayag pour traiter tous les aspects de la composition assistée et de la musicologie compositionnelle incluant la syntaxe sonore et l'interaction. Aujourd'hui, OpenMusic, distribué par le Forum de l'Ircam, touche un millier d'utilisateurs dans le monde et est enseigné dans un grand nombre d'institution (Ircam, UCSD, U. Columbia, U. Berkeley, U. Sussex, U. of Limerick, Keele University, Amsterdam Conservatory, Sibelius Academy). Les travaux autour d'OpenMusic ont suscité plusieurs travaux académiques dont trois thèses : celle de Carlos Agon et Jean Bresson à l'Ircam et celle de Fabio Selviaforita à l'Université de Bologne.

ML-Maquette avec Jean Bresson et Carlos Agon, une version pédagogique d'OpenMusic pour le jeune public des lycées et collèges, 2005-2007. Ce logiciel, commandé par l'Education Nationale est utilisé par les professeurs d'éducation musicale.

Sympathos avec Georges Bloch, un système de simulation de la résonance sympathique pour les claviers électroniques, 1995. Brevet : Process for the simulation of sympathetic resonances on an electronic keyboard instrument, Patent number: 5854438, Filing date: Apr 8, 1997, Issue date: Dec 29, 1998, Inventors: Gerard Assayag, Georges Bloch, Assignee: France Telecom, U.S. Classification 84625; 84627; 84633 International Classification G10H 1057; G10H 146; G10H 700.

NexoCAAD, 1988, simulation en acoustique des salles, modeleur 3-D, modèles de propagation acoustique, aide à la conception de salles de concert et de systèmes de sonorisation. 1988.

AMS, 1989, système de mesures et de traitement du signal temps réel pour l'acoustique des salles

HyperXplorer, système de contrôle du dispositif mécanique d'acoustique variable avec commande vocale pour le studio d'Eric Serra (1989).

Boards, 1987-1990, réalisé en free-lance, reprend les principes de **Crime** et ajoute la notion de génération automatique de template d'interface graphique spécialisé inspiré de l'Interface Builder de Jean Marie Hullot (qui devait plus tard devenir le cœur de l'IHM de la NeXt machine)

Crime, 1983-1987, est le premier environnement d'informatique symbolique au service du compositeur mis en place à l'Ircam. Basé sur des principes de calcul formel et des descriptions algébriques des structures musicales, il pouvait aussi imprimer des partitions en notation musicale traditionnelle.

Radian d'Art, 1980, programme d'IA générateur de poèmes selon les règles de la phonologie générative. Lauréat des concours Art et Informatique et Concours Micro de l'Agence de l'informatique et Ministère de la recherche 1980.

19. Bibliographie du domaine

Abadi, M. and Cardelli, L., «A Theory of Primitive Objects: Untyped and First-order Systems», *Proceedings of Theoretical Aspects of Computer Software*, Springer Verlag, 1994.

Acotto, E., Andreatta, M., « Représentations mentales musicales et représentations mathématiques de la musique », *InCognito, Cahiers Romains de Sciences Cognitives*, 2008

Agon, C., Bresson, J., « Mixing Time Representations in a Programmable Score Editor », *Sound and Music Computing Conference*, Lefkada, 2007

Agon, C., « Mixing Visual Programs and Music Notation », *Perspectives of Mathematical and Computer-Aided Music Theory*, ed. Lluís-Puebla Emilio, Mazzola Guerino, Noll Thomas. (epOs Music, Osnabrück), 2003

Ahn, Y.K., Agon, C., Andreatta, M., « "Structures Ia pour deux pianos" by Boulez : towards creative analysis using OpenMusic and Rubato », *Mathematics and Computation in Music '07*, Berlin, 2007, pp. 234-238

Ahn, Y.K., « Aspects théoriques et computationnels de l'analyse transformationnelle », *Mémoire de Master II Atiam, Paris 6 / IRCAM*, 2006

Allauzen C., Crochemore M., Raffinot M., « Factor oracle: a new structure for pattern matching », in *Proceedings of SOFSEM'99, Theory and Practice of Informatics*, J. Pavelka, G. Tel and M. Bartosek ed., Milovy, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin 1999, n°1725 pp 291-306

Allen, J. F. «Maintaining Knowledge About Temporal Intervals», *Communications of the ACM* 1983, 26(11):832-843.

Allombert, A., Desainte-Catherine, M., « Interactive scores: A model for specifying temporal relations between interactive and static events », *Journal of New Music Research*, Décembre 2005, vol. 34, n° 4, pp. 361-374

Amiot, E., Agon, C., Andreatta, M., « Autosimilar melodies and their implementation in OpenMusic », *Proceedings Sound and Music Computing SMC'07*, Lefkada, 2007

Amiot, E., Noll, T., Andreatta, M., Agon, C., « Fourier Oracles for Computer-Aided Improvisation », *ICMC 2006*, New Orleans, 2006

Andreatta, M., « La musique mise en algèbre », *Pour la science*, Novembre 2008, n° 373

Andreatta, M., « La géométrie d'un Prélude : des nouvelles représentations géométriques des structures musicales éclairent une pièce de Chopin qui ne finit pas de séduire les mélomanes. », *Pour la science*, Novembre 2006, n° 349

Andreatta, M., « A computer-aided exploration of the transformational process in Xenakis'Nomos Alpha », AMS/MAA Joint Mathematics Meeting, vol. 25 n.1, Phoenix AZ, 2004, pp. 588-58

Andreatta, M., « On group-theoretical methods applied to music: some compositional and implementational aspects », Perspectives of Mathematical and Computer-Aided Music Theory, ed. Lluís-Puebla Emilio, Mazzola Guerino, Noll Thomas. (epOs Music, Osnabrück), 2003

Andreatta M., Agon C., "Elementi di teoria matematica della musica e problema della classificazione di canoni regolari complementari di categoria massimale" Il Monocordo, Vol. 7/8, 2000.

Andreatta, M., Agon, C., « A tutorial on algebraic-oriented paradigmatic music analysis and composition in OpenMusic visual programming language », Fourth MUSICNETWORK Open Workshop – Integration of Music in Multimedia Applications, Barcelone, 2004

Andreatta, M., Agon, C., « Combining audio and symbolic representation in OpenMusic visual programming language : a multimedia model of Iannis Xenakis' Nomos Alpha », Fourth MUSICNETWORK Open Workshop – Integration of Music in Multimedia Applications, Barcelone, 2004

Andreatta, M., Agon, C., Amiot, E., « Tiling problems in music composition: Theory and Implementation », proceedings ICMC'02, Göteborg, 2002

Andreatta, M., Bardez, J.M., Rahn, J., « Around Set Theory : a French-American Musicological Meeting », Around Set Theory (Ircam/Delatour France, Sampzon), 2008

Andreatta, M., Schaub, S., « Une introduction à la Set Theory: les concepts à la base des théories d'Allen Forte et de David Lewin », Musurgia, 2003, vol. X, n° 1, pp. 73-92

Andreatta, M., Vuza, D. T. «On some properties of periodic sequences in Anatol Vieru's Modal Theory», Tatra Mt. Math. Publ. 23, 2001, pp. 1-15, Haluska Ed. Bratislava.

Apple. Opendoc technical summary. Technical documentation, Apple Computer, 1994.

Ayari, M. «De la théorie musicale à l'art de l'improvisation : analyse de performances et modélisation musicale», DELATOUR-France, collection « Culture et Cognition Musicales», Paris, 2005.

Ayari, M. & Lartillot, O., «Perception et Modélisation Cognitive». In Revue Intellectica, collection «Musique et Cognition», sous la direction de Anne Sedes, Université de Paris VIII, Septembre 2006.

- Ayari, M. & McAdams, S., «L'influence du savoir culturel sur la perception et la compréhension musicales», In D. F. Madurell (éd.), *Les pratiques d'écoute individuelles. Série "Conférences et séminaires", N° 20*, Université Paris Sorbonne, Paris IV. 2005, pp 63-83.
- Ayari, M. & McAdams, S., «Quelles catégories et quels filtres culturels sont à l'œuvre dans nos écoutes ?», In Ayari, M (éd.), *De la théorie musicale à l'art de l'improvisation : analyse de performances et modélisation musicale*. DELATOUR-France, collection « Culture et Cognition Musicales », Paris, 2005.
- Baisnée, P. and J. Barrière, M. Dalbavia, J. Duthen, M. Lindburg, Y. Potard, K. Saariaho. 1988. «ESQUISSE: A Compositional Environment», In *Proceedings of the International Computer Music Conference 1988*. pp. 108-118.
- Balaban, M. «The Music Structures Approach to Knowledge Representation for Music Processing», *Computer Music Journal*, 20(2), pp. 96-111, 1996.
- Balaban, M., K. Ebcioglu and O. Laske, eds. *Understanding Music with AI*. Cambridge, Massachusetts and Menlo Park, California: The MIT Press and AAAI Press. 1992
- Ballesta, P. «Contraintes et objets: clefs de voûte d'un outil d'aide à la composition» in *Informatique Musicale, Recherche et Applications*. Chemillier and Pacht Eds, Hermes, 1998.
- Balpe, J.P., «Hypertexte et interactivité». In *Hypertextes et hypermédias*, n°1/1997 (mars 1997), éditions Hermès.
- Bancquart, A., Andreatta, M., Agon, C., « Microtonal Composition », *The OM Composer's Book 2*, ed. Jean Bresson, Carlos Agon, Gérard Assayag. (Ircam/Delatour France, Sampzon), 2008, pp. 279-302
- Barbaud, P., «La musique discipline scientifique», Paris, Dunod, 1968
- Barendregt, H.P., «The Lambda Calculus: Its Syntax and Semantics, *Studies in Logic and the Foundations of Mathematics*», 103 (Revised edition ed.), North Holland, Amsterdam 1983.
- Baratè, A., Haus, G., Ludovico, L., «Petri Nets Applicability to Music Analysis and Composition», *Proceedings of the International Computer Music Conference '07 (ICMC 2007)*, Copenhagen 2007, pp. 97-100
- Bejerano G. and G. Yona, «Variations on Probabilistic Suffix Trees: Statistical Modeling and Prediction of Protein Families», *Bioinformatics*, vol. 17, 2001, pp. 23-43.
- Belinda Thom, «BoB: an interactive improvisational music companion», *Proceedings of the Fourth International Conference on Autonomous Agents*, ACM Press, Barcelona, Catalonia, Spain 2000, pp. 309-316

Bestougeff H., et G. Ligozat. Outils logiques pour le traitement du temps. Masson, Paris 1989

Biles, J.A. «Genjam in perspective: A tentative taxonomy for genetic algorithm music and art systems». *Leonardo* 36(1), 43–45 (2003)

Boynton, L., Lavoie, P., Orlarey, Y., Rueda, C., Wessel D., «MIDI-LISP, a LISP-based music programming environment for the Macintosh» *Proceedings of the International Computer Music Conference 1986, Computer Music Association ICMA, San Francisco, CA*. Pages 183-186

Bresson, J., Agon, C., « Musical Representation of Sound in Computer-Aided Composition : a Visual Programming Framework », *Journal of New Music Research*, 2007, vol. 36, n° 4, pp. 251-266

Bresson, J., Agon, C., « Scores, Programs, and Time Representation : the Sheet Object in OpenMusic », *Computer Music Journal*, 2008, vol. 32, n° 4, pp. 31-47

Bresson, J., Agon, C., « SDIF sound description data representation and manipulation in computer assisted composition », *International Computer Music Conference ICMC'04, Miami, 2004*

Bresson, J., Agon, C., « Temporal Control over Sound Synthesis Processes », *Sound and Music Computing SMC'06, Marseille, 2006*

Bresson, J., Stroppa, M., Agon, C., « Generation and Representation of Data and Events for the Control of Sound Synthesis », *Sound and Music Computing Conference SMC'07, Lefkada, 2007*

Bresson, J., Stroppa, M., Agon, C., « Symbolic Control of Sound Synthesis in Computer-Assisted Composition », *International Computer Music Conference ICMC'05, Barcelona, 2005*

Bresson, J., « Processus compositionnels et opérateurs musicaux dans ML-Maquette - Les outils de traitement du signal », *Journées d'Informatique Musicale JIM'07, Lyon, 2007*

Bresson, J., « Sound Processing in OpenMusic », *9th International Conference on Digital Audio Effects - DAFx-06, Montréal, 2006*, pp. 325-330

Briot, J.P., «Des Objets aux Acteurs - 1982-1989: 7 Ans de Réflexion», *Habilitation à Diriger des Recherches, Rapport de Recherche 89-68, LITP, UPMC, Paris, France, Septembre 1989*

Briot J.P., and Cointe, P., «The ObjVLisp model: Definition of a uniform, reflexive and extensible object-oriented language», In Benedict du Boulay, Daniel Hogg, and Luc Steels, editors, *European Conference on Artificial Intelligence (ECAI'86)*, Brighton, U.K., *Advances in Artificial Intelligence-II*, North-Holland, pages 225-232, 1987.

- Camurri, A «On the role of artificial intelligence in music research», *Interface*, 1990, 19(2-3), 219-248.
- Chailloux, J., Devin, M., Hullot, «Le_Lisp, a Portable and Efficient Lisp System», Rapport technique Inria RT0319, Inria Rocquencourt 1984.
- Chazal, G. «Le miroir automate», Seyssel: Ed. Champ Vallon 1995
- Chemillier M., «Analysis and Computer Reconstruction of a Musical Fragment of György Ligeti's *Melodien*», *Muzica* 6 (2) Bucharest (1995) 34-48
- Chemillier M., «Structure et méthode algébriques en informatique musicale», Thèse Univ. Paris 7, 1989
- Chemillier, M., Truchet, C., « Computation of words satisfying the "rhythmic oddity property" », *Information Processing Letters*, 2003, n° 86, pp. 255-261
- Chemillier, M., « Periodic musical sequences and Lyndon words », *Soft Computing*, Octobre 2004, vol. 9, n° 8, pp. 611-616
- Chemillier, M., « Représentations musicales et représentations mathématiques », *L'Homme*, 2004, n° 171, pp. 267-284
- Chemillier, M., « Toward a formal study of jazz chord sequences generated by Steedman's grammar », *Soft Computing*, 2004, vol. 9, n° 8, pp. 617-622
- Chew, Elaine (2008). *Math & Music - The Perfect Match*. *OR/MS Today*, June Issue, Volume 35, Number 3, pp.26-31
- Chew, Elaine. «Regards on Two Regards by Messiaen: Post-tonal Music Segmentation Using Pitch Context Distances in the Spiral Array», Gerard Assayag (guest ed.): *Special Issue on Sound and Music Computing*, *Journal of New Music Research*, 2005, 34(4), pp.341-354
- Chouvel, J.M., «Représentation harmonique hexagonale toroïde» *Musimediane*, revue en ligne 2005, N° 1, www.musimediane.com
- Chouvel, J.M., Bresson, J., Agon, C., « L'analyse musicale différentielle : principes, représentation et application à la structuration des entités musicales pour la musique électroacoustique », *EMS07 - Electroacoustic Music Studies*, Leicester, 2007
- Codognet P. and D. Diaz. «Yet Another Local Search Method for Constraint Solving», *Proceeding of the AAAI Symposium «Using Uncertainty in Computation»*, T. Walsh, C. Gomes (Eds.), AAAI Press, Cape Cod October 2001
- Conklin D. and I. Witten, "Multiple Viewpoint Systems for Music Prediction," *Interface*, 1995, vol. 24, pp. 51-73
- Conklin, D. « Music Generation from Statistical Models », *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, Aberystwyth, Wales, 2003, pp. 30- 35

- Cont, A., «Modeling Musical Anticipation: From the time of music to the music of time», PhD thesis, University of Paris 6 and University of California in San Diego, October 2008
- Cope, D. «Computer Models of Musical Creativity», The MIT Press 2005
- Cope, D. «Experiments in Musical Intelligence», Madison, WI: A-R Editions, 1996.
- Courtot, F. «CARLA : Knowledge acquisition and induction for computer assisted composition». Rapport de recherche, Ircam, 1990
- Courtot, F. «Entre le décomposé et l'incomposable», Les cahiers del'Ircam : La composition assistée par ordinateur, Paris juin 1993, 1(3)
- de Cheveigné, A., Kawahara, H. «YIN, a fundamental frequency estimator for speech and music», J. Acoust. Soc. Am. 2002, 111, 1917-1930.
- Dannenberg, R.B. «Machinetongues XIX : Nyquist, a language for composition and sound synthesis», Computer Music Journal, 1997, 21(3), 50–60.
- Dannenberg, R.B., Desain, P. & Honing, H. «Programming language design for music». In C. Roads, S.T. Pope, A. Piccialli & G. DePoli (Eds.), Musical signal processing. Lisse: Swets and Zeitlinger 1997
- Dannenberg, R.B. «Music Representation Issues, Techniques, and Systems», Computer Music Journal, 1993, Volume 17, Number 3.
- Delerue, O., Agon, C. «OpenMusic + MusicSpace = OpenSpace», Proceedings JIM'99, CeMaMu, Issy-les-moulineaux. 1999.
- Diaz Frias J., Rueda, C., Valencia F. D. «A calculus for concurrent processes with constraints» CLEI 97, Congreso Latino Americano de Informatica, Valparaiso, Chili, Aout 1997
- Diomidis D. Spinellis. «Programming Paradigms as Object Structuring Mechanism for Multiparadigm Programming». PhD thesis, Science, Technology and Medicine, London, 1994.
- Dixon, S., «Automatic extraction of tempo and beat from expressive performances.», Journal of New Music Research, 2001 30 :39–58
- Dubnov, S., McAdams, S., and Reynolds, R. «Structural and affective aspects of music from statistical audio signal analysis», Special Topic Section on Computational Analysis of Style. J. Am. Soc. Inf. Sci. Technol. 57, 11 (Sep. 2006), 1526-1536
- Duthen, J. and M. Laurson. 1990. «A Compositional Environment based on Pre-FORM II, PatchWork, and Esquisse», In Proceedings of the International Computer Music 1990, pp. 228-230
- Ebcioğlu, K. «An expert system for harmonizing four-part chorales», Machine models of music, MIT Press, Cambridge, MA, 1992

- Eckel, G., González-Arroyo, R. «Musically Salient Control Abstractions for Sound Synthesis». Proceedings of the 1994 International Computer Music Conference Aarhus, 1994.
- Eckel, G. «La maîtrise de la synthèse sonore» In Les cahiers de l'Ircam 2 : La synthèse sonore. Paris: Ircam – CentrePompidou, 1993
- Eckel, G. «Manipulation of sound signals based on graphical representation. A musical point of view», In Proceedings of the 1992 International Workshop on Models and Representations of Musical Signals, Capri, Italia, 1992
- Ehresmann, A., Vanbreemeersch, J.P., «Memory Evolutive Systems: Hierarchy, Emergence, Cognition», Studies in Multidisciplinarity Vol. 4, Elsevier, 2007 (389 pages).
- Ehrgott M., Gandibleux, X. «A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization», OR Spektrum 22 (2000), 425–460.
- Estrada, J., Gil, J., «Música y teoría de grupos finitos (3 variables booleanas), with an English resume», 1984, Mexique, IIE, UNAM, 221 pp.
- Farbood, M., Kaufman, H., Jennings. «Composing with Hyperscore: An intuitive interface for visualizing musical structure», Proceedings of the 2007 International Computer Music Conference.
- Feder, M., N. Merhav, and M. Gutman, «Universal Prediction of Individual Sequences», IEEE Trans. Information Theory, 1992, 38(4)1258-1270
- Forte, A., «The Structure of Atonal Music», Yale University Press, 1973
- François, A. R. J., E. Chew, and C. D. Thurmond «Visual Feedback in Performer-Machine Interaction for Musical Improvisation», In Proceedings of International Conference on New Instruments for Musical Expression, New York, June, 2007
- Gilbert, E., « Polynômes cyclotomiques, canons mosaïques et rythmes k-asymétriques », Mémoire de Master II Atiam, Université Paris VI, 2007.
- Godet, J.B., « Grammaires de substitution harmonique dans un improvisateur automatique », Mémoire de DEA Atiam, Paris 6, Ircam 2004.
- Gomez, E., Klapuri, A., Meudic, B., « Melody Description and Extraction in the Context of Music Content Processing », Journal of New Music Research (JNMR), 2003, vol. 32, n° 1
- Gouyon, F., Meudic, B., « Towards rhythmic content processing of musical signals --- Fostering complementary approaches », Journal of New Music Research (JNMR), 2003, vol. 32, n° 1
- Hamel, K. «NoteAbility, A Comprehensive Music Notation System», Proceedings of the 1998 International Computer Music Conference, 1998, pp. 506–509.

- Henz, M., G. Smolka and J. Wurtz, «Object-Oriented Concurrent Constraint Programming in Oz», in: *Principles and Practice of Constraint Programming*, V. Saraswat and H. Montanari Eds, MIT Press, 1995
- Hoos, H.H., K. A. Hamel, K. Renz, J. Kilian. «The GUIDO Music Notation Format : A Novel Approach for Adequately Representing Score-level Music», *ICMC'98 Proceedings*, 1998, p.451-454.
- Hornel, D., Ragg. T., «A Connectionist Model for the Evolution of Styles of Harmonization», In: *Proceedings of the 1996 International Conference on Music Perception and Cognition*. Montreal, 1996.
- Hullot, J.M., «Alcyone, la boîte à outils objets», *Rapport Technique Inria N° 60*, RT0060, 1985
- Huron, D. «Sweet Anticipation: Music and the Psychology of Expectation». MIT Press, Cambridge (2006)
- Jaskiewicz, A. «Comparison of Local Search-Based Metaheuristics on the Multiple Objective Knapsack Problem», *Foundations of Computing and Design Sciences* 26 (2001), no. 1, 99–120.
- Junod, J., « Etude combinatoire et informatique du caractère diatonique des échelles à sept notes », mémoire de Master II Atiam, UPMC - Paris VI, 2008.
- Kuuskankare, M., and M. Laurson «Expressive Notation Package», *Computer Music Journal* 2006, 30(4):67–79.
- Lartillot, O., «An Adaptive Multi-Parametric and Redundancy-Filtering Approach for Motivic Pattern Discovery», *Journal of New Music Research*, 34-4, pp. 375-393, Taylor & Francis, 2006.
- Lartillot, O. «A Musical Pattern Discovery System Founded on a Modeling of Listening Strategies». *Comput. Music J.* 28, 3 (Sep. 2004), 53-67
- Lartillot, O. & Ayari, M., «Motivic Pattern Extraction in Music : And Application to the Study of Tunisian Modal Music », In *Advance in end-user data mining techniques*, Co-organisé par ARIMA (Revue africaine en informatique et mathématiques appliquées) et SACJ (South African Computer Journal) octobre 2005.
- Lartillot, O., « Discovering Musical Patterns through Perceptive Heuristics. », *ISMIR 2003*, Baltimore, 2003
- Lartillot, O., « Perception-Based Advanced Description of Abstract Musical Content. », *Digital Media Processing for Multimedia Interactive Services*, ed. Ebroul Izquierdo. (World Scientific Press, Singapour), 2003, pp. 320-323
- Lartillot, O., « Analyser sans réduire : un modèle cognitif d'induction d'analogies », *Peut-on parler d'art avec les outils de la science ?*, ed. Jean-Marc Chouvel, Fabien Lévy. (Ircam - L'Harmattan, Paris), 2002, pp. 195-217

- Lartillot, O., « Generalized Musical Pattern Discovery by Analogy from Local Viewpoints. », *Discovery Science*, ed. Steffen Lange, Ken Satoh, Carl H. Smith. (Springer-Verlag, Heidelberg), 2002, pp. 382-389
- Lartillot, O., « Integrating Pattern-Matching into an Analogy-Oriented Pattern Discovery Framework. », *ISMIR: 3rd International Conference on Music Information Retrieval*, Paris, 2002, pp. 269-270
- Lartillot, O., « Musical Analysis by Computer Following Cognitive Model of Induction of Analogies. », *ICMC: International Computer Music Conference*, Göteborg, 2002, pp. 20-27
- Lartillot, O., Saint-James, E., « Automating Motivic Analysis through the Application of Perceptual Rules. », *Computing in Musicology*, ed. Walter B. Hewlett, Eleanor Selfridge-Field. (MIT Press, Cambridge, MA), 2003
- Laske, O. «Composition theory in Koenig's project one and project two», *Computer Music Journal*, 1981, 5(4),54–65.
- Laurson, M., and M. Kuuskankare, «PWGL: A Novel Visual Language based on Common Lisp, CLOS and OpenGL». *Proceedings of International Computer Music Conference, San Francisco, California 2002*, pp.142–145.
- Laurson M., «PATCHWORK: A Visual Programming Language and some Musical Applications». Helsinki : Sibelius Academy, Helsinki, 1996.
- Laurson M. «PWConstraints». In Haus G. and Pighi I. (eds), *X Colloquio di Informatica Musicale*, 332-335, Associazione di Informatica Musicale Italiana, . Milano 1993.
- Lefebvre, A., Lecroq, T. «Computing repeated factors with a factor oracle», in: L. Brankovic, J. Ryan (Eds.), *Proceedings of the 11th Australasian Workshop On Combinatorial Algorithms*, Hunter Valley, Australia, 2000, pp. 145-158.
- Lelong G. «Le stade de l'intégration» in *Les Cahiers de l'IRCAM n°6 "Musique:texte"*, 1994
- Leman, M., Ed. «Music, Gestalt, and Computing - Studies in Cognitive and Systematic Musicology», Springer-Verlag, London, UK 1997
- Leman,M. «Symbolic and sub symbolic description of music». In G. Haus (Ed.), *Music Processing*. Oxford: OxfordUniversityPress, 1993
- Longuet-Higgins, H.C. (1979). «The perception of music». *Proceedings of the Royal Society B* 205: 307–322
- Malt, M. « Les mathématiques et la Composition assistée par Ordinateur (Concepts, Outils, Modèles)», thèse de l'EHESS. Paris, 2001.
- Marsden, A. «Representing Musical Time: A Temporal-Logic Approach», Swets & Zeitlinger. Lisse, Netherlands 2000

- Maxwell, J. T.; Ornstein, S. M. «Mockingbird: a composer's amanuensis». *BYTE Magazine*. 1984 January.
- Mazzola, G., « La vérité du beau dans la musique », ed. Jean-Michel Bardez et Moreno Andreatta, Delatour, Sampzon, 2007
- Mazzola, G., Andreatta, M., « Diagrams, gestures and formulae in music », *Journal of Mathematics and Music*, Mars 2007, vol. 1, n° 1, pp. 23-46
- Mazzola, G., Andreatta, M. «From a Categorical Point of View: K-nets as Limit Denotators». *Perspectives of New Music*, 2006 volume 44, number 2
- McAdams, S. 1999 «Perspectives on the Contribution of Timbre to Musical Structure.» *Comput. Music J.* 23, 3 (Sep. 1999), 85-102
- McAdams, S., Battier, M., (Eds.). «Perception and Creation of a Contemporary Musical Work: The Angel of Death by Roger Reynolds», Paris, Ircam-Centre Pompidou [Multimedia], 2004
- Mesnage, M., Riotte, A., « Modélisation informatique de partitions, analyse et composition assistée », *Cahiers de l'Ircam n° 3* (Ircam - Centre Georges Pompidou, Paris), 1993
- Meudic, B., « A causal algorithm for beat-tracking », 2nd conference on understanding and creating music , Caserta, Italy, 2002
- Meudic, B., St-James, E., « Automatic extraction of approximate repetitions in polyphonic MIDI files based on perceptive criteria », *Computer Music Modeling and Retrieval*, ed. Uffe Kock Wiil. (Springer, Berlin), 2003, pp. 124-142
- Milner, R., J. Parrow and D. Walker, *A Calculus of Mobile Processes*, *Journal of Information and Computation*, 1992
- Miranda, E. R., Biles, J. A., «Evolutionary Computer Music», Springer: 2007, 259p.
- Mogens N., and Valencia, F. D. « Temporal Concurrent Constraint Programming: Applications and Behavior ». Chapter in W. Brauer, H. Ehrig, J. Karhumäki, A. Salomaa (Eds.), *Formal and Natural Computing*, Springer-Verlag 2002, pp. 298-324.
- Murail, T., « Spectres et lutins », in *Darmstädter Beiträge zur Neuen Musik n°19*, Schott, 1982, p. 24-34, [réédité dans *L'Ircam – une pensée musicale*, Editions des Archives Contemporaines, 1984]
- Orlarey, Y., Fober, D. & Letz, S. «Syntactical and semantical aspects of Faust» *Soft Computing*, 2004, 8(9), 623–632
- Orlarey, Y., Fober D., Letz, S. «Elody : a Java+MidiShare based Music Composition Environment», *Proceedings of the ICMC 97*, Thessaloniki, 1997.
- Pachet, F., Ramalho G., Carrive, J. «Representing Temporal Musical Objects and Reasoning in the MusES System», *Journal of New Music Research* 1996, N° 3

- Pachet, F., «Musical Interaction with Style». *Journal of New Music Research*, 2003. 32(3): p. 333-341.
- Pachet, P. « Interacting with a Musical Learning System: The Continuator », *Proc. ICMAI 2002*, Springer-Verlag, 2002, pp. 119-132.
- Pachet, P., Roy, P. « Musical harmonization with constraints : a survey», *Constraints Journal*, Kluwer Publisher 2001, 6(1)7-19
- Paepcke, A. «Object-Oriented Programming - The CLOS Perspective». MIT Press, 1993.
- Pearce, M., Conklin, D., Wiggins, G. «Methods for combining statistical models of music». In: Wiil, U.K. (ed.) *CMMR 2004*. LNCS, vol. 3310, pp. 295–312. Springer, Heidelberg (2005)
- Plans Casal, D., Morelli, D. «Remembering the Future : and Overview of Co-Evolution in Musical Improvisation», in *proceedings of International Computer Music Conference*, Copenhagen, August 2007, pp.200-206
- Puckette, M., «Max at seventeen» *Computer Music Journal* 2002, 26(4): pp. 31-43.
- Puckette, M. «Combining event and signal processing in the MAX graphical programming environment», *Computer Music Journal*, 1991, 15(3), 68–77
- Ramalho, G., Rolland, P.-Y., & Ganascia, J.-G. «An Artificially Intelligent Jazz Performer», *Journal of New Music research*, Swets & Zeitlinger: Amsterdam 1999, 28(2). pp. 105-129.
- Read, G., «Music Notation: A Manual of Modern Practice», *Crescendo Publishing*, 1964; revised 1972 *Taplinger Publishing Company*, 1979 (2nd edition)
- Riotte, A. (Ed.), *Musrgia* Vol. VI/1 : La génétique des oeuvres, 1999
- Riotte, A., Mesnage, M., «Formalismes et modèles musicaux», 2 volumes, *Collection Sciences/Musique*, Ircam Delatour France, 2006
- Risset, J.C., «Musique, calcul secret ?» *Critique*, 1977, n° 395, pp. 414-429
- Roads, C. «The Computer Music Tutorial», Cambridge, MA: MITPress, 1996
- Rodet, X. «The Diphone program: new features, new synthesis methods, and experience of musical use», *Proc. of the ICMC 1997*, Thessaloniki, Hella 1997, pp. 418-21.
- Rodet, X., Cointe, P. «FORMES: Composition and scheduling of processes». In C. Roads, editor, *The Music Machine*, Cambridge Massachusetts, 1984. MIT Press.
- Ron, D., Y. Singer, and N. Tishby, "The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length," *Machine Learning*, vol. 25, 1996, pp. 117-149.

- Rowe, R. «Interactive Music Systems: Machine Listening and Composing», MIT Press 1992
- Rowe, R. «Machine Musicianship», MIT Press 2004
- Rueda C., Bonnet A. « Un langage visuel basé sur les contraintes pour la composition musicale ». in Pachet F. et Chemiller M. (eds), *Recherches et applications en informatique musicale*, Hermes, Collection Informatique Musicale, Paris, 1998.
- Rueda, C., Tamura, G., Quesada, L. O., «The Visual Model of Cordial», CLEI 97, Congreso Latino Americano de Informatica, Valparaiso, Chili, Aout 1997
- Reenskaug, T., «Model, View», Controllers» internal note, Xerox Park Palo Alto, Dec 1979, av. at <http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf>
- Schaub, S., « Akrata, for 16 winds by Iannis Xenakis: Analysis », International Symposium Iannis Xenakis, Athènes, 2005, pp. 138-149
- Schottstaedt, B. «Common Music Notation». In : *Beyond MIDI, The handbook of musical codes*. Eleanor Selfridge-Field ed. MIT press, Cambridge Masssachusetts, 1997.
- Seleborg, C., « Interaction temps-réel/ temps différé », Mémoire de DEA Atiam, Paris 6, Ircam 2004.
- Siskind J. M. and McAllester D. «Nondeterministic Lisp as a Substrait for Constraint Logic Programming». AAI-93, 1993, pages 133-138.
- Smith, L. «SCORE Reference Manual». Half Moon Bay, CA: Passport, 1987
- Steedman M.J., «A Generative Grammar for Jazz Chord Sequences», *Music Perception* 1984 2:1 52-77
- Steele, G. L. «Common LISP The language», second edition. Digital Press, USA, 1998.
- Stickney, K. P. «Computer Tools for Engraving-Quality Music Notation», *The Proceedings of the AES 5th International Conference, Music and Digital Technology*. Audio Engineering Society 1987.
- Stroppa, M., Lemouton, S., Agon, C., « OmChroma ; vers une formalisation compositionnelle des processus de synthèse sonore », JIM 2002, Marseille, 2002
- Stroppa, M., and Duthen, J. «Une representation de structures temporelles par synchronisation de pivots», *Colloque Musique et Assistance Informatique*, Marseille, France: Laboratoire Musique et Informatique de Marseille (MIM), 1990, pp. 305-322.
- Terhardt, E., Stoll, G., & Seewann, M. «Algorithm for extraction of pitch and pitch salience from complex tonal signals», *Journal of the Acoustical Society of America*, 1982, 71, 679-688.

- Tack, G., *Constraint Propagation -- Models, Techniques, Implementation*, Doctoral Dissertation, Saarland University, Jan 2009
- Taube, H. «Common Music Reference Manual», University of Illinois, Illinois 2002.
- Taube, H. «Common Music: A Compositional Language in Common Lisp and CLOS», In *Proceedings of the International Computer Music Conference*. San Francisco: International Computer Music Association, 1989, pp. 316-319.
- Taube, H. «SAL: A Simple Algorithmic Language in Common Music», In *Proceedings of the International Computer Music Conference*. San Francisco: International Computer Music Association, 2007, 1: 121-124.
- Tardieu, D., Rodet, X. «An Instrument Timbre Model For Computer Aided Orchestration», WASPAA (New Paltz, NY, USA), Octobre 2007.
- Triviño-Rodríguez J.L. And R. Morales-Bueno, « Using Multiattribute Prediction Suffix Graphs To Predict And Generate Music », *Computer Music Journal*, Spring 2001, Volume 25 No. 3, pp. 62-79
- Truchet, C., Codognet, P., « Musical Constraint Satisfaction Problems Solved with Adaptive Search », *Soft Computing*, Septembre 2004, vol. 8, n° 9, pp. 633-640
- Truchet, C., « Some Constraint Satisfaction Problems in Computer Assisted Composition », *Perspectives of Mathematical and Computer-Aided Music Theory*, ed. Lluís-Puebla Emilio, Mazzola Guerino, Noll Thomas. (epOs Music, Osnabrück), 2003
- Tsang, C., Aitken, M. «Harmonizing music as a discipline of constraint logic programming» *ICMC*, 1991.
- Vasconcelos, V.T., «A process-calculus approach to typed concurrent objects», PhD Thesis, Keio University 1994.
- Vuza, D.T., *Supplementary Sets and Regular Canons I*, *Perspectives of New Music*, 29(2). 1991.
- Wright, M. & Freed, «A. Open Sound Control: a new protocol for communicating with sound synthesizers», In *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997
- Xenakis I. «Musiques formelles», Stock Musique, Paris, 1981.
- Zicarelli, D. «M and Jam Factory», *Computer Music J.*, vol. 11, no. 4, 1987, pp. 13-29
- Ziv, J., Lempel, A. «Compression of Individual Sequences via Variable Rate Coding», *IEEE Trans. Information Theory*, vol. 24, no. 5, 1978, pp. 530-536
- Ziv, J., Lempel, A., «A universal algorithm for sequential data compression», *IEEE Trans. Inform. Theory* 23 (3) (1977) 337-343.
- Zuber, H., « Vers une arithmétique des rythmes ? », *Mémoire de Master I de mathématiques*, ENS Cachan, Ircam 2005