



HAL
open science

Les logiciels d'aide à la composition musicale

Dominique Fober, Stéphane Letz

► **To cite this version:**

Dominique Fober, Stéphane Letz. Les logiciels d'aide à la composition musicale. Rencontres Musicales Pluridisciplinaires, 1999, Lyon, France. pp.13-19. hal-02158893

HAL Id: hal-02158893

<https://hal.archives-ouvertes.fr/hal-02158893>

Submitted on 18 Jun 2019

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

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

Les logiciels d'aide à la composition musicale

Dominique Fober - Stéphane Letz

[fober ; letz]@rd.grame.fr

Grame - Laboratoire de recherche

9, rue du Garet BP 1185

69202 LYON Cedex 01

Introduction

Le XXème siècle aura vu le champ d'action du domaine musical prodigieusement élargi, à la fois par l'apparition de nouveaux moyens techniques de production du son, que par l'apparition de nouveaux modes d'investigation de la pensée musicale. Celle-ci s'affranchit des procédés classiques de composition pour explorer une grande variété et une grande richesse de modèles de pensée qui naissent notamment de la confrontation et de l'échange avec des domaines scientifiques eux aussi en pleine mutation.

C'est au début du siècle, dans les années 20 que le domaine musical s'ouvre à l'art des sons avec notamment les travaux et concerts des bruitistes Marinetti et Russolo. C'est à la fin des années 40 que cet art nouveau prendra forme sous le nom de musique concrète, sous l'impulsion de Pierre Schaeffer et de Pierre Henry. Apparaît alors un nouveau mode de pensée musicale qui réside plus dans la nature des phénomènes sonores mis en oeuvre que dans leur organisation.

En parallèle de cette évolution, l'art de la composition subit des transformations profondes, liées à la remise en cause des formes et du langage musical. Le dodécaphonisme dont les principes sont exposés dès les années 20 par Joseph Matthias Hauer, marquera le déclin de l'univers tonal et la naissance de nouvelles techniques d'écriture dont les principes seront définis par Schönberg dans le cadre de la musique sérielle, et dont l'extension à toutes les composantes du son, dans un formalisme rigoureux, verra son apogée dans les années 50.

C'est à la fin des années 50 que naîtra l'informatique musicale. Cette discipline créera progressivement des outils qui vont s'inscrire en prolongement de ces évolutions, mais vont surtout leur ouvrir des perspectives beaucoup plus vastes tout en modifiant radicalement les moyens d'action

de la pensée musicale. En amplifiant ses capacités de calcul, l'ordinateur permet au compositeur d'élaborer des oeuvres dont le temps de réalisation échapperait aux possibilités humaines. En lui permettant d'expérimenter et d'explorer rapidement de nombreuses hypothèses, il offre à la pensée musicale une liberté et une mobilité nouvelles.

Les apports de l'informatique dans le domaine musical vont cependant dépasser largement le seul phénomène d'amplification de la pensée. Il faudra pour que l'ordinateur puisse traiter de musique, inventer de nouveaux systèmes de représentation et d'écriture, prenant en compte tant le matériau que les processus de la composition, rendant ces derniers explicitement accessibles à la pensée musicale.

Les nouveaux espaces de la pensée musicale

La première utilisation de l'ordinateur à des fins de composition musicale date de 1957. Lejarden Hiller et Leonard Isaacson créent cette année là le "Quatuor à cordes N°4" plus connu sous le nom de "Suite Illiac", en hommage à l'ordinateur, l'ILLIAC I, qui permit de calculer l'oeuvre.

En parallèle de ses études de chimie à Princeton, Hiller étudia la composition avec Milton Babbitt et Roger Sessions. Il devint membre de la faculté de chimie de l'Université de l'Illinois en 1952, date à laquelle l'ILLIAC I fut mis en service. Il utilisa son expérience de l'ILLIAC au développement de règles pour la composition musicale. Il argumente sa démarche en considérant que «...le processus de composition musicale peut être caractérisé comme l'extraction d'ordre d'une multitude chaotique de possibilités disponibles...». Le matériau de base pour la *Suite Illiac* est généré en grande quantité avec l'utilisation de l'algorithme de Monte-Carlo [Hiller 1981]. Cet algorithme produit des nombres

aléatoires qui sont ensuite codifiés et associés à différents paramètres musicaux tels que la hauteur, l'intensité ou encore des modes de jeu. Ce matériel est enfin soumis à un ensemble de règles compositionnelles implémentées en utilisant la technique des chaînes de Markov, selon lesquelles la probabilité d'un état (dans notre cas, la valeur d'un paramètre musical) est strictement déterminée par les états antérieurs. Ce modèle des chaînes de Markov est développé sous le nom de bigrammes, trigrammes ou n -grammes, selon que l'on calcule la probabilité d'un état donné en fonction des deux, trois ou n états précédents.

Le résultat de la démarche de Hiller est plus important du point de vue de la recherche que sur le plan de l'esthétique musicale. A propos de la *Suite Illiac*, il considère que «...son objectif en composant de la musique à l'aide d'un ordinateur n'est pas de réaliser une unité esthétique, mais de fournir une évaluation des techniques qui permettraient une nouvelle prise en compte cette esthétique...» [Moore 1996].

En ayant recours aux mathématiques et à l'informatique pour les mettre en œuvre, Hiller ouvre le champ de la composition automatique. A cet égard, les travaux de Pierre Barbaud occupent une place importante. Il choisit dès les années 50, d'introduire «la pensée mathématique et les méthodes qui en découlent dans la composition musicale». Avec Roger Blanchard et Janine Charbonnier, il fondera le groupe de musique algorithmique de Paris. Les machines Bull prêteront leur concours à leurs expérimentations qui produiront en 1958, dix programmes sous le nom symbolique d'Algol.

De même que pour Hiller, Barbaud reprend les théories aléatoires en postulant que la musique oscille entre l'ordre et le désordre. Il ira cependant beaucoup plus loin dans la manière de traduire ses idées dans un langage propre à l'ordinateur, incluant dans sa démarche une formalisation cohérente de la théorie musicale, invitant à repenser de façon nouvelle et générale, l'ensemble des connaissances de l'harmonie, de la fugue du contrepoint etc.

Ses travaux reposent sur la théorie des groupes et la théorie des graphes qui servent de base à l'analyse du langage tonal. L'œuvre musicale est le résultat de l'application de règles à des matériaux exprimés sous la forme d'ensembles. Les règles sont définies à partir d'automates finis ou de matrices stochastiques. Son œuvre «7!» (1960) est reconnue comme la première œuvre réalisée par ordinateur en France.

La contribution de Iannis Xenakis dans le domaine de l'informatique musicale est tout aussi déterminante. Architecte de formation, il collabora avec Le Corbusier de 1947 à 1959 pour se consacrer

ensuite entièrement à la musique. C'est en 1954, alors qu'il travaillait à la conception du pavillon Philips, qu'il eut l'idée de projeter les lignes définissant les surfaces courbes du pavillon sur du papier à musique, pour y figurer des glissandi de violons ou d'autres instruments. Le résultat, *Metastasis*, est une œuvre d'une grande originalité, tant du point de vue de la conception que du résultat sonore. Dans *Pithoprakta* (1956), pour quarante-six instruments à cordes et un trombone, chaque note, est assimilée à une molécule extrêmement mobile, telle la molécule d'un gaz contenu à l'intérieur d'une enceinte à une température et sous une pression données : c'est l'application aux sons de la formule de Maxwell-Boltzmann.

En ces années 50, Xenakis tente de reconstituer dans sa musique des événements sonores analogues aux phénomènes naturels sous forme de "nuages" (bruits de foule, pluie) régis par les lois des grands nombres, de Poisson, de Gauss, de Maxwell. Il trouvera dans le calcul de probabilité, les outils conceptuels lui permettant d'organiser le hasard. C'est sous le nom de musique stochastique qu'il formalisera sa démarche, applicable à tous les paramètres du son [Xenakis 1963]. L'utilisation de l'ordinateur, à partir de 1958, lui apportera un gain de temps précieux : réussissant à convaincre la compagnie IBM France de la validité de son projet, il obtiendra l'autorisation de se servir gracieusement du monument électronique qu'était l'ordinateur 7090. Il l'utilisera notamment pour calculer la série des ST/4, ST/10 et ST/48, pièces écrites respectivement pour 4, 10 et 48 instruments.

Citons encore la démarche de Michel Philippot qui avec le *Septuor (Mm/d2) in memoriam Isaac Newton* (1977) utilise une technique d'écriture qui s'apparente à un calcul de simulation de mécanique céleste.

Plus proches de nous, Jérôme Dorival et Yann Orlarey utilisent des automates cellulaires pour concevoir *Divin Chaos* (1990), Pierre Alain Jaffrennou développe des techniques de composition par processus avec le *Solitaire bulgare* (1993). Des terminologies nouvelles apparaissent (telles que "*musiques fractales*") dénotant la multiplication de démarches qui vont toutes dans le sens de l'appropriation de modèles provenant de domaines scientifiques ou technologiques très divers.

Les outils d'exploration de ces nouveaux espaces de la pensée musicale seront fournis par l'informatique et nous nous pencherons plus particulièrement sur les langages d'aide à la composition dans la mesure où ils présentent le plus de généralité en laissant au musicien toute liberté pour élaborer son propre langage musical. Dans cette perspective, l'utilisation de l'ordinateur à des fins de composition suppose de disposer :

- d'une représentation des données musicales qui permette leur manipulation symbolique par l'ordinateur,
- d'un langage de programmation qui opérera sur ces données par le biais de leur représentation symbolique,
- d'outils d'écriture, qui permettront à l'utilisateur, à la fois de composer sa musique et de programmer la machine.

Il est important de rappeler ici l'importance des systèmes de représentation [Dannenberg 1993], [Wiggins & al. 1993]. En particulier, l'organisation du temps occupe une position stratégique dans la problématique de la représentation musicale [Honing 1993]. Rappelons également que la relation entre écriture et composition, qui se situent dans une même dynamique, participent étroitement du même processus [Nicolas 1991]. En ce sens il faut considérer l'apport des outils informatiques d'aide à la composition musicale, non comme simples auxiliaires du musicien, mais comme étant appelés à faire partie intégrante du processus d'écriture.

Quelques repères historiques

MUSICOMP (Music Simulator Interpreter for Compositional Procedures) est communément cité comme le premier langage informatique d'aide à la composition musicale. Il a été conçu par Lejaren Hiller et Robert Baker en 1963 et résulte des travaux préliminaires pour la composition de "*Computer Cantata*". Dans les faits il s'agit d'une librairie de sous-routines écrites en FORTRAN, s'appuyant sur les paradigmes de générateurs, modificateurs et sélecteurs. Les routines génératrices sont basées sur les chaînes de Markov, les modificateurs appliquent des transformations sérielles et géométriques, et les sélecteurs effectuent un filtrage des matériaux précédents à partir de règles tirées principalement de l'harmonie classique. Le terme de langage ici peut être compris au sens de langage musical et illustre la démarche particulière de Hiller.

Au Pays-Bas, entre 1965 et 1970, Gottfried Michael Koenig réalise les programmes Project I et Project II. Project I calcule son résultat musical à partir d'une base de données à laquelle le musicien fournit une contribution limitée sous la forme de durées, tempi et nombre d'événements, ainsi qu'un nombre aléatoire initialisant le processus de calcul [Laske 1981]. La forme générale de l'oeuvre, donnée a priori, n'est pas accessible au compositeur. Une partie l'acte de création est donc confiée à l'ordinateur. C'est l'époque où celui-ci est considéré comme un "cerveau électronique".

Les travaux de compositeurs tels que Charles Ames ou David Cope se situent dans la même perspective puisqu'ils visent à établir des domaines de paramètres musicaux sur lesquels ils pourront appliquer des structures de contrôle automatique, qui produiront la structure générale de l'oeuvre [Cope 1987]. Ces recherches sont toujours vivaces de nos jours. Elles contribuent au domaine de l'intelligence artificielle, grâce notamment à des emprunts à la linguistique.

En 1972, poursuivant sa démarche théorique, Xenakis crée le C.E.M.A.Mu. (Centre d'Études de Mathématiques et d'Automatique Musicales). Il sera le premier à proposer d'unifier synthèse sonore, représentation et composition musicale en concevant l'UPIC (Unité Polyagogique du CEMAMu) à partir de 1975. Il s'agit d'un système de synthèse sonore contrôlé par une large tablette graphique, proposant au compositeur de dessiner sa musique et permettant de calculer le résultat sonore de la page composée [Raczinski, Marino 1988], [Marino & al. 1990]. La démarche est généreuse puisqu'elle vise à mettre à la portée de tous, les moyens de réaliser des intentions musicales sans avoir à passer par l'apprentissage du solfège ni de l'informatique. Elle met en évidence les dimensions nouvelles de l'écriture musicale induites par l'utilisation de l'ordinateur et aura des prolongements jusqu'à nos jours.

L'apparition de l'ordinateur personnel et l'établissement de la norme MIDI (Musical Instrument Digital Interface) en 1983, vont permettre la multiplication de recherches qui fonderont l'ère "moderne" des outils d'aide à la composition. La norme MIDI a été développée par l'industrie des synthétiseurs avec pour motivation première de permettre la communication entre instruments électroniques et le partage d'informations telles que le contrôle gestuel [Loy 1985]. Il s'agit donc à la fois d'un protocole de communication et d'un système de représentation non pas du son, mais du geste du musicien jouant d'un instrument MIDI. Ce standard va très rapidement remporter un succès inattendu et suscitera le développement commercial de nombreux logiciels musicaux, ouvrant des perspectives nouvelles à la recherche, notamment à travers la confrontation à un public élargi. En une quinzaine d'année, plus d'une centaine de langages musicaux verront le jour, témoignant tant du dynamisme de l'informatique musicale que de la difficulté à construire des outils ayant une bonne persistance dans le temps. Ces langages relèvent des systèmes de représentation de la musique et des langages de programmation dédiés à la synthèse sonore ou la composition musicale. Pour une bonne approche des systèmes de description de la musique, on peut se reporter notamment à [BeMI 1997]. On trouvera dans [Loy

1989], une synthèse très fournie de ces débuts de l'informatique musicale.

Les langages d'aide à la composition musicale

Il n'est pas possible de citer ici tous les travaux existants. Dans une perspective historique, nous retiendrons tout d'abord ceux qui fondèrent le domaine. Parmi les plus importants figurent :

- HMSL (Hierarchical Music Specification Language) (1980) développé par Phil Burk, Larry Polansky et David Rosenboom à Mills College en Californie [Polansky, Rosenboom 1985]. C'est un ensemble d'extensions au langage Forth dont les concepts fondamentaux sont tirés de la perception et des théories de la cognition.
- PLA (1983), développé au CCRMA, Université de Stanford par Schottstaedt [Schottstaedt 1989], est un langage interprété écrit en SAIL, un dialecte de la famille Algol alors très utilisé au CCRMA. PLA va inclure progressivement des modèles de langages très différents, tels que Lisp, SmallTalk ou encore SCORE qui fournira les capacités de notation musicale.
- Arctic (1984), développé par Roger Dannenberg [Dannenberg & al. 1986], explore la capacité des modèles de programmation fonctionnelle à exprimer les problèmes de contrôle temps réel propres au domaine musical. Sa particularité est de permettre la déclaration de relations entre les fonctions du temps plutôt que de stipuler des procédures qui vont produire ces relations. Le déroulement du temps est ainsi rendu explicite dans le langage.
- Common Music (1989), développé par Heinrich Taube au CCRMA [Taube 1989]. C'est un environnement de composition orienté objet développé en Common Lisp qui permet de transformer des structures musicales de haut niveau en une grande variété de protocoles de contrôle du son (MIDI, CSound, Common Lisp Music...) ou de rendu graphique (Common Music Notation). Common Music fera l'objet d'extensions notamment à travers *Stella* [Taube 1992], un système textuel de représentation permettant à la fois une description procédurale et littérale d'une partition musicale, ainsi que *Capella*, une interface graphique au langage de programmation.

Les contributions de l'IRCAM dans ce domaine s'inscrivent dans une continuité remarquable :

- FORMES (1983) développé par Xavier Rodet et Pierre Cointe [Rodet, Cointe 1984]. Il s'agit d'un langage de programmation orienté objet, doté de caractéristiques originales, dédié à la composition musicale et à l'ordonnancement temporel d'objets. Une interface de programmation graphique,

PreFORM, sera ensuite définie [Boynton & al. 1986a].

- CRIME (1984) développé par Gérard Assayag est le premier environnement permettant de manipuler des formalismes musicaux avec des résultats sous forme de partition en notation traditionnelle.
- Esquisse (1987) développé sous l'impulsion de Jean Baptiste Barrière [Baisnée & al. 1988], est une application implémentée sur PreFORM, qui consiste en une boîte à outils permettant de faire des opérations musicales. Esquisse a fourni un apport important dans la classification des fonctions par rapport au type de connaissance musicale qu'elles représentent.
- CARLA (1989), écrit en Prolog par Francis Courtot, met en oeuvre les paradigmes de la programmation logique dans un environnement graphique. C'est un des premiers logiciels adapté à la visualisation des processus de composition.
- Patchwork (1989), conçu par Laurson, Duthen et Rueda [Laurson, Duthen 1989], consiste en une interface graphique pour Common Lisp dont le développement sera repris et amélioré jusqu'en 1995. Le modèle de programmation consiste à connecter des boîtes représentant des fonctions Lisp et possédant des entrées (les arguments) et une sortie (le résultat).
- OpenMusic (1996), conçu par Gérard Assayag et Carlos Agon [Assayag, Agon 1996], [Agon 1998], étend les concepts de programmation visuelle développés avec Patchwork, dans un environnement totalement graphique, s'appuyant sur le langage Common Lisp et dans un paradigme de programmation objet. OpenMusic intègre un système de représentation du temps d'une grande souplesse et fournit sous la forme de *maquettes*, des outils originaux permettant de spécifier à la fois l'ordonnancement des objets de la composition ainsi que leurs relations.

Avec la même constance, le laboratoire de recherche de Grame a également proposé plusieurs environnements d'aide à la composition musicale :

- Midi Logo (1984), développé par Yann Orlarey [Orlarey 1986] qui s'appuie sur le langage LOGO pour fournir un environnement de programmation reposant sur la métaphore du magnétophone multipistes pour l'organisation du matériel musical et proposant des primitives originales de manipulation des "têtes de lecture" de ces pistes.
- Midi-Lisp (1986), développé en collaboration avec l'IRCAM et Act Informatique [Boynton & al. 1986b]
- CLCE (1991), développé par Yann Orlarey, Bertrand Merlier et Stéphane Letz. C'est un environnement de composition basé sur Common Lisp, qui permet notamment d'aborder de manière homogène, les problèmes de composition et

d'interaction, par un ensemble d'outils utilisables indifféremment en temps réel et en temps différé.

- Elody (1997), développé par Orlarey, Letz et Fober [Orlarey & al. 1997] C'est un langage fonctionnel à manipulation visuelle directe. La spécificité d'Elody tient à la grande homogénéité de tous les éléments du langage. En particulier, tous les objets ont une dimension temporelle y compris les fonctions qui peuvent être mixées ou mises en séquence comme n'importe quel autre matériel musical. L'approche développée avec Elody présente de plus un intérêt général pour l'ensemble des langages d'aide à la composition puisqu'elle montre que l'on peut aisément transformer un langage purement descriptif en langage de programmation par la simple adjonction de l'abstraction et de l'application du lambda calcul.

Les travaux de Daniel Oppenheim méritent également d'être mentionnés pour leur originalité. DMIX (1989) est un environnement d'aide à la composition où les notions de transformation résident dans des "modifiers" qui consistent en données associées à des opérations mathématiques [Oppenheim 1989]. Ces données peuvent être dérivées d'objets musicaux, de telle sorte qu'il est alors aisé de transformer un objet musical en "modifier" pour l'appliquer à un autre objet musical. Cette démarche est très proche des modèles de programmation fonctionnelle et confère une grande homogénéité aux différents éléments du langage. Elle sera reprise en partie dans la conception du logiciel CyberBand (1996) qui développera une contribution particulière au domaine des langages pour utilisateurs finaux [Oppenheim & al. 1997].

Dans la perspective des études menées dans les années 80 par Seymour Papert [Papert 1980] sur la cognition de la programmation, l'informatique musicale explorera également le domaine des langages pour utilisateurs non-programmeurs. Ce sont par exemple les travaux de Desain et Honing qui avec LOCO, proposeront un environnement de composition basé sur le langage LOGO [Desain, Honing 1986].

L'exploration de nouveaux espaces sonores.

C'est probablement dans le domaine du son que se sont opérées les mutations les plus profondes de la musique du XXème siècle. La synthèse, l'analyse du son et toutes les techniques développées autour du traitement du signal, ont ouvert l'accès à des espaces sonores considérés comme sans limites. L'informatique musicale consacrera la majorité de ses travaux aux outils d'exploration de ces espaces.

C'est en 1957, l'année où naquit la *Suite Illiac*, que commence l'histoire de ces outils. Max Mathews dirige alors dans le cadre des Bell Telephone Labs, un programme de recherche visant à développer des techniques d'analyse et de synthèse sonore. Il conçoit le programme MUSIC I qui lui permit de calculer les premiers sons de synthèse numérique. MUSIC I fut le premier d'une série de programmes qui aboutiront au langage MUSIC V (1968). Ces travaux seront repris et prolongés jusqu'à nos jours par Richard Moore (CMusic - 1985) et Barry Vercoe (CSound - 1986).

Dans les années 60 et toujours dans le cadre des laboratoires Bell, les travaux de Jean Claude Risset marquèrent le début d'une longue lignée de recherches visant à mieux comprendre la structure interne du son musical [Pierce 1984].

Fondée sur ces techniques et sur cette connaissance nouvelle de la matière acoustique, l'informatique musicale va alors créer des logiciels qui permettront aux musiciens de composer avec la structure intime du phénomène sonore. Deux directions complémentaires vont se développer : les outils et les méthodes pour la synthèse sonore, et les outils pour la "sculpture" sonore, s'appliquant à des sons concrets, permettant de les modifier et de les modeler au désirs du compositeur. A travers le nombre, l'informatique tendra à unifier la pensée musicale en lui permettant d'embrasser toute la "tessiture" de son champ d'action, désormais élargie à la "micro-composition".

S'il n'est pas dans notre propos de détailler ici l'ensemble des travaux touchant au domaine du son, leur importance et leur impact sur la composition musicale font qu'on ne peut non plus de les passer sous silence. Les ouvrages traitant de ce domaine ne manquent pas, pour une vue historique ou synthétique de ces travaux, on pourra se référer à [Roads 1980], [Loy 1989] et [Moore 1996].

Conclusion

L'utilisation de l'ordinateur à des fins de création musicale s'inscrit en toute logique dans la généralisation de l'emploi de l'outil informatique. Elle résulte également d'un détournement : l'ordinateur n'a pas été conçu pour l'art, ce qui pose un certain nombre de questions. Il est par exemple, particulièrement adapté à résoudre des problèmes bien spécifiés alors qu'à l'inverse, la démarche artistique consiste souvent à explorer toutes les ramifications possibles d'une idée en germe. Comment dès lors, concilier formalisation préalable et attitude prospective ? En déplaçant le problème de la conception à celui du cheminement de l'esprit permettant d'aboutir à l'oeuvre, en

proposant des langages de programmation comme moyens d'expression de la pensée artistique, l'informatique musicale fera preuve de beaucoup d'invention et de diversité pour proposer des solutions originales, adaptées à la création musicale.

Témoin de la diversité : le choix d'un paradigme de programmation par exemple; il est porteur d'implications profondes sur la manière d'exprimer ses idées; un langage particulier suscitera des expressions ou des manières de structurer sa pensée qui ne seraient pas favorisées dans un autre langage. L'informatique musicale aura exploré tous les paradigmes de programmation existants, proposant des environnements relevant aussi bien de la programmation objet que de la programmation logique ou fonctionnelle.

L'ordinateur impose également des systèmes de représentation discrets, numériques : l'informatique musicale mettra cette contrainte à son profit pour revisiter toute la théorie de la musique, à la lumière des langages descriptifs qu'elle élaborera pour permettre la manipulation symbolique du matériau musical.

Enfin, c'est très tôt et avec constance que seront pris en compte les aspects cognitifs de la programmation, tant pour ce qui est de l'écriture que de son adaptation à la pensée musicale. En effet, les langages informatique traditionnels sont régis par des grammaires et des syntaxes très éloignées du langage musical, leur apprentissage nécessite un temps et un effort qui sont souvent hors de portée des compositeurs, les représentations textuelles sont mal adaptées à la représentation musicale... Les environnements d'aide à la composition ont déployé de nombreuses stratégies pour réduire le fossé entre les deux domaines, proposant des interfaces graphiques, des modes de programmation plus intuitifs, des représentations en notation traditionnelle ou encore des langages monotype où les erreurs de syntaxes n'existent plus. De ce point de vue, les travaux cités précédemment témoignent des capacités d'adaptation de l'informatique musicale.

Mentionnons également toutes les recherches réalisées parallèlement dans les domaines de l'analyse et de l'intelligence artificielle : elles auront permis au monde musical de porter un regard nouveau sur son passé.

Les outils résultants de l'ensemble de ces travaux auront prodigieusement élargi le champ de la pensée musicale. Mais plus encore, celle-ci se verra enrichie de nouvelles dimensions : les langages d'aide à la composition permettent d'accéder au processus mis en oeuvre, de les représenter et de les soumettre au calcul ; ainsi objectivée dans l'écriture, la pensée musicale peut alors devenir

matériau de la composition. De cette dimension nouvelle naîtra notamment la musique interactive.

Le champ de l'écriture musicale est immense et nul doute que l'informatique y tient désormais une place importante. Rendons hommage pour finir à tous ceux, trop nombreux pour être cités, qui ont contribué à fonder ce domaine par leur travaux et leur persévérance.

Références

[Agon 1998] Carlos Agon, *OpenMusic: un langage visuel pour la composition musicale assistée par ordinateur*. Thèse de doctorat de l'Université de Paris 6. IRCAM, 1998

[Assayag, Agon 1996] Gérard Assayag, Carlos Agon, *OpenMusic Architecture*. Proceedings of the ICMC 1996, Computer Music Association, San Francisco, pp.339-340

[Baisnée & al. 1988] Pierre-Francois Baisnée, Jean-Baptiste Barriere, Marc-André Dalbavie, *ESQUISSE: A Compositional Environment*. Proceedings of the ICMC 1988, Computer Music Association, San Francisco, pp.108-118

[Battier 1995] Marc Battier, *Entre l'idée et l'oeuvre : Parcours de l'informatique musicale*. in Esthétique des arts médiatiques, Presses de l'Université du Québec, 1995

[BeMI 1997] *Beyond MIDI. The Handbook of Musical Codes*. Edited by Eleanor Selfridge-Field, MIT Press, 1997

[Boynton & al. 1986a] Boynton, Duthen, Potard, Rodet, *Adding a Graphical User Interface to FORMES*. Proceedings of the ICMC 1986, Computer Music Association, San Francisco, pp.105-108

[Boynton & al. 1986b] Boynton, Lavoie, Orlarey, Rueda, Wessel, *MIDI-LISP, a LISP-based music programming environment for the Macintosh*. Proceedings of the ICMC 1986, Computer Music Association, San Francisco, pp.183-186

[Cope 1987] David Cope, *An Expert System for Computer-assisted Composition*. Computer Music Journal, Vol 11, N°4, Winter 1987, MIT Press

[Dannenberg & al. 1986] Roger B. Dannenberg, Paul McAvinney, Dean Rubine, *Arctic: A Functional Language for Real-Time Systems*. Computer Music Journal, Vol 10, N°4, Winter 1986, MIT Press

[Dannenberg 1993] Roger B. Dannenberg, *Music Representation Issues, Techniques, and Systems*. Computer Music Journal, Vol 17, N°3, Fall 1993, MIT Press

- [Desain, Honing 1986] Peter Desain, Henkjan Honing, *LOCO: Composition Microworlds in LOGO*. Proceedings of the ICMC 1986, Computer Music Association, San Francisco, pp.109-118
- [Honing 1993] Henkjan Honing, *Issues in the representation of time and structure in music*. Contemporary Music Review, 9, 1993, pp.221-239
- [Hiller 1981] Lejaren Hiller, *Composing with Computers : A Progress Report*. Computer Music Journal, Vol. 5, N°4, Winter 1981, MIT Press
- [Laske 1981] Otto Laske, *Composition Theory in Koenig's Project One and Project Two*. Computer Music Journal, Vol 5, N°4, Winter 1981, MIT Press
- [Laurson, Duthen 1989] Mikael Laurson, Jacques Duthen, *PATCHWORK: a Graphic Language in PREFORM*. Proceedings of the ICMC 1989, Computer Music Association, San Francisco, pp.172-175
- [Loy 1985] Gareth Loy, *Musicians Make a Standard : The MIDI Phenomenon*. Computer Music Journal, Vol 9, N°4, Winter 1985, MIT Press
- [Loy 1989] Gareth Loy, *Composing with Computers - a Survey*. in Current Directions in Computer Music Research. edited by Max Mathews and John Pierce, 1989, MIT Press, pp.291-396
- [Marino & al. 1990] Gérard Marino, Jean Michel Raczinski, Marie Hélène Serra, *The new UPIC system*. Proceedings of the ICMC 1990, Computer Music Association, San Francisco, pp.249-252.
- [Moore 1996] F. Richard Moore, *Dreams of Computer Music - Then and Now*. Computer Music Journal, Vol 20, N°1, Spring 1996, MIT Press
- [Nicolas 1991] François Nicolas, *Huit thèses sur l'écriture musicale*. Revue Analyse musicale N°24, Paris, Juin 1991
- [Oppenheim 1989] Daniel V. Oppenheim, *DMIX: An Environment for Composition*. Proceedings of the ICMC 1989, Computer Music Association, San Francisco, pp.226-233
- [Oppenheim & al. 1997] Oppenheim, Wright, Jameson, Pazel, Fuhrer, *CyberBand: A "Hands-On" Music Composition Program*. Proceedings of the ICMC 1997, Computer Music Association, San Francisco, pp.383-386
- [Orlarey 1986] Yann Orlarey, *MLOGO: A MIDI Composing Environment for the Apple IIe*. Proceedings of the ICMC 1986, Computer Music Association, San Francisco, pp.211-213
- [Orlarey & al. 1997] Yann Orlarey, Stéphane Letz, Dominique Fober, *L'environnement de composition musicale Elody*. Actes des Journées d'Informatique Musicale, JIM'97, 1997, Grame - Lyon, pp.122-136
- [Papert 1980] Seymour Papert, *Mindstorms*. Harvester Press, 1980
- [Pierce 1984] John R. Pierce, *Le son musical*. Collection "L'univers des sciences", Belin, 1984
- [Polansky, Rosenboom 1985] Larry Polansky, David Rosenboom, *HMSL A Real-Time Environment for Formal, Perceptual and Compositional Experimentation*. Proceedings of the ICMC 1985 pp.243-250 Computer Music Association, San Francisco,
- [Raczinski, Marino 1988] Jean Michel Raczinski, Gérard Marino, *A real time synthesis unit*. Proceedings of the ICMC 1988, Computer Music Association, San Francisco, pp.90-99.
- [Roads 1980] Curtis Roads, *Interview with Max Mathews*. Computer Music Journal, Vol 4, N°4, Winter 1980, MIT Press
- [Rodet, Cointe 1984] Xavier Rodet, Pierre Cointe, *FORMES: Composition and Scheduling of Processes*. Computer Music Journal, Vol 8, N°3, pp.32-50, MIT Press
- [Schottstaedt 1989] William Schottstaedt, *A Computer Music Language*. in Current Directions in Computer Music Research. edited by Max Mathews and John Pierce, 1989, MIT Press, pp.215-224
- [Taube 1989] Heinrich Taube, *Common Music, A Compositionnal Language in Common Lisp and CLOS*. Proceedings of the ICMC 1989, Computer Music Association, San Francisco, pp.316-319
- [Taube 1992] Heinrich Taube, *Stella: persistent score representation in Common Music*. Proceedings of the ICMC 1992, Computer Music Association, San Francisco, pp.101-105
- [Wiggins & al. 1993] Geraint Wiggins, Eduardo Miranda, Alan Smaill, Mitch Harris, *A Framework for the Evaluation of Music Representation Systems*. Computer Music Journal, Vol 17, N°3, Fall 1993, MIT Press
- [Xenakis 1963] Iannis Xenakis, *Musiques Formelles*. n° spécial 253-254, La Revue Musicale, 1963, Stock, 1981