



HAL
open science

Annotation automatique et (re)composition : poétique et technique d'un flux de travaux par base de données

Alessandro Ratoci

► **To cite this version:**

Alessandro Ratoci. Annotation automatique et (re)composition : poétique et technique d'un flux de travaux par base de données. Journées d'Informatique Musicale, JIM2020, Oct 2020, Strasbourg, France. hal-03353949

HAL Id: hal-03353949

<https://hal.science/hal-03353949>

Submitted on 24 Sep 2021

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.



Distributed under a Creative Commons Attribution 4.0 International License

ANNOTATION AUTOMATIQUE ET (RE)COMPOSITION : POÉTIQUE ET TECHNIQUE D'UN FLUX DE TRAVAUX PAR BASE DE DONNÉES.

Alessandro Ratoci

Sorbonne Université - STMS Lab, IRCAM
Haute école de musique Vaud Valais, Fribourg,
HES-SO Haute école spécialisée de Suisse occidentale
IRMAS Institute de Recherche en Musique et Arts de la Scène
alessandro.ratoci@hemu-cl.ch

RÉSUMÉ

L'objectif de cette recherche est d'établir un flux de travaux pour la composition musicale à partir de la recombinaison, à différents échelles, d'unités sémantiques constituées par des fragments extraits (ou générés) à partir des musiques préexistantes. Dans l'articles sont décrits les motivations esthétiques et aussi les méthodes et les outils qui ont permis la réalisation d'une œuvre pour guitare électrique et électronique basé sur l'extraction des éléments rythmiques par des musiques du groupe *noise rock* « Sonic Youth ». Ces outils s'appuient sur le concept de base de données, implémenté dans le langage SQLite, et offrent un support pour la réflexion autour du matériel précompositionnel et pour le processus formalisé de composition.

1. INTRODUCTION

Tous les actes créatifs, y compris la composition musicale, ne s'accomplissent qu'à travers des contraintes. Igor Stravinsky, dans sa *Poétique Musicale*, résume de manière emblématique le problème de la liberté dans le processus de composition, nous rappelant comment le créateur opère à travers l'auto-imposition des contraintes une réduction de l'espace des solutions possibles. Uniquement lorsque le parcours qui lie le champ du potentiel à sa réalisation devient suffisamment tortueux, le compositeur se libère de la sensation que l'acte décisionnel est arbitraire, et avec le sentiment que chaque élément produit par ses choix est « nécessaire » à la cohérence de l'œuvre, il se livre finalement à l'expression subjective et personnelle « libéré des chaînes qui enchaînent l'esprit » :

« My freedom thus consists in my moving about within the narrow frame that I have assigned myself for each one of my undertakings. I shall go even further: my freedom will be so much the greater and more meaningful the more narrowly I limit my field of action and the more I surround myself with obstacles. Whatever diminishes constraints, diminishes strength. The more constraints one imposes, the more one frees one's self of *the chains that shackle the spirit* » [16].

La fixation de ces contraintes se produit en première lieu au niveau du matériel précompositionnel [15]. Soit qu'il s'agisse d'esquisses rythmiques et mélodiques, de « réservoirs » de hauteurs, de séquences de timbres ou même des bibliothèques de sons concrets ou d'algorithmes génératifs, les façons de produire ce matériel de départ sont nombreuses. Elles vont de l'improvisation instrumentale aux opérations logico-mathématiques jusqu'à l'aléa pur (John Cage et ses dés) en passant par la réutilisation, dans une certaine mesure, d'éléments de musiques préexistantes.

Cet article, qui découle de ma pratique de compositeur, décrit les outils et les flux de travaux lorsque les matériaux précompositionnels symboliques (hauteurs, rythmes) sont extraits à partir d'une source musicale préexistante, sous forme d'enregistrement discographique. La nouvelle composition se produit donc par élaboration et recombinaisons des éléments morphologiques d'une musique « autre » qui est en hétérogène pour genre et style. Cette opération présente des traits communs avec la pratique du *remix* des musiques électroniques « ludiques », mais dans un sens plus élargi et dans un contexte « mixte » (qui passe donc aussi par l'écriture instrumentale). Dans le cas spécifique, il s'agit d'une pièce pour guitare électrique et électronique en temps réel « Confused Idols & Sexy Killers » écrit en 2018 pour Luca Nostro sur commande du Ravenna Festival. Le travail est entièrement basé sur la réélaboration des deux premières EP « Confusion is Sex » and « Kill yr Idols » du band *noise rock* new-yorkaise Sonic Youth.

Dans cet article sera pris en considération le rôle de l'analyse et de l'annotation automatique dans mon esthétique et les outils permettant de réaliser ces deux étapes du processus créatif. Successivement sera détaillé le flux de travail par base de données qui a permis la génération du matériel précompositionnel et enfin sera montré quelque exemple de comme le matériel a fait son chemin à travers la phase de composition instrumentale et électronique.

2. L'ANNOTATION AUTOMATIQUE

De plus en plus fréquemment dans mon travail de composition, le processus d'extraction des éléments symboliques à partir des musiques préexistantes est délégué à l'ordinateur. À travers les techniques d'annotation automatique du signal audio [11][7] qui font partie du champ plus large du *Music Information Retrieval* (MIR) il est possible d'obtenir plusieurs informations relatives à l'articulation du temps, du timbre et du contenu mélodique et harmonique. Dans la réalisation de « Confused Idols & Sexy Killers », l'analyse a été principalement conduite sur le contenu rythmique (*onset detection*) [3] et les informations obtenues sont utilisées pour la structuration du rythme et des durées de la macro-forme. Cette segmentation constitue aussi le point de départ de l'analyse du contenu harmonique par moyennement des partiels (*chord-seq analysis*) [4] qui génère le matériel à la base de l'organisation mélodique et harmonique de la pièce.

À la différence des utilisations « canoniques » des outils de MIR qui sont liés au marché de la musique numérique (catalogage en ligne, outils pour les Dj, etc...) et qui sont destinés à assister ou substituer l'écoute, en fournissant des résultats les plus proches possible de la capacité de discernement de l'oreille humaine, ces outils peuvent se libérer de toute obligation de « haute-fidélité » dans les mains du compositeur. Chaque algorithme d'analyse appliqué au matériel de départ ne produit pas qu'une seule solution, mais un champ de solutions possibles, qui est généré à travers la multiplication cartésienne de ses paramètres. Plus la granularité des paramètres est fine, plus leur combinaison donne naissance à un espace continu de solutions, toutes potentiellement acceptables comme matériel pré-compositionnel. Ce qui est considéré comme erreur dans le *benchmarking* des techniques d'analyse peut donc devenir une première ressource pour l'élaboration du matériel de départ, détournant le MIR vers le rôle d'agent génératif de substances musicales « autres ».

La difficulté du travail est, dans cette première étape, d'organiser, trier et filtrer une quantité considérable de données et d'en reconnaître le potentiel musical. Le compositeur doit donner un sens à cette profusion de données qui se présentent souvent comme des listes « muettes » de durées ou des agrégats chaotiques qui correspondent aux partiels d'un son polyphonique. Pour accomplir cette tâche, le compositeur a donc besoin d'instruments qui puissent lui donner accès immédiat à ces matériels « bruts » avec la rapidité et dans une disposition compatible avec le flux de son imagination. En règle générale le compositeur ne possède ni le temps ni la nécessité d'utiliser tout l'ensemble des données générées, mais il doit plutôt parcourir, « naviguer » dans cet espace de possibilités (comme Xenakis le faisait remarquer au sujet des expériences de composition assistée tout court) [18]. Il doit pouvoir sauvegarder, évaluer, labéliser, modifier et transcrire ces éléments au fur et à mesure de leur découverte. La découverte et l'élaboration sont complémentaires, la chaîne des trans-

formations doit se produire sans empêcher un retour en arrière selon une logique d'exploration par « branches ». Le paradigme est donc l'organisation principale de cette façon de procéder, la « base des données » l'espace qui l'organise et l'interrogation, la duplication et l'*undo* ses fonctions « primitives ». Il s'agit donc d'adopter plutôt une posture proche de celle du naturaliste qui observe une population extrêmement diversifiée plutôt que celle du créateur qui interroge sa propre inspiration devant une page blanche. Dans son travail « Music across music : towards a corpus-based, interactive computer-aided composition » [10] le compositeur et chercheur italien Daniele Ghisi explique clairement : les corrélations et les oppositions entre deux paradigmes conceptuels qui peuvent s'appliquer à la composition musicale assistée par ordinateur, celle de « l'invention », et de la « découverte ».

3. LA BASE DE DONNÉES ET L'ENVIRONNEMENT INFORMATIQUE

Dans la même recherche, Ghisi développe une librairie pour l'environnement Max nommé *dada* qui fournit des instruments utiles pour son approche à la composition musicale, parmi ceux-ci se trouve un objet spécifique *dada.base* qui permet l'implémentation d'une base de données de type SQLite dans un environnement de CAO.

Le compositeur peut réunir dans un seul espace opérationnel les différentes étapes de l'évolution des matériels. Cette démarche permet une exploration par « branches », ou plusieurs chaînes de transformations peuvent être conduites en parallèle avec des *sets* de paramètres différents et des résultats aisément comparables entre eux. Il s'agit à mon avis d'un aspect assez important, car il permet aux compositeurs qui ont généralement une connaissance limitée des mathématiques subjacente aux algorithmes de MIR, de comprendre la fonction que chaque paramètre apporte grâce à l'observation continue et massive de la corrélation avec son résultat musicale.

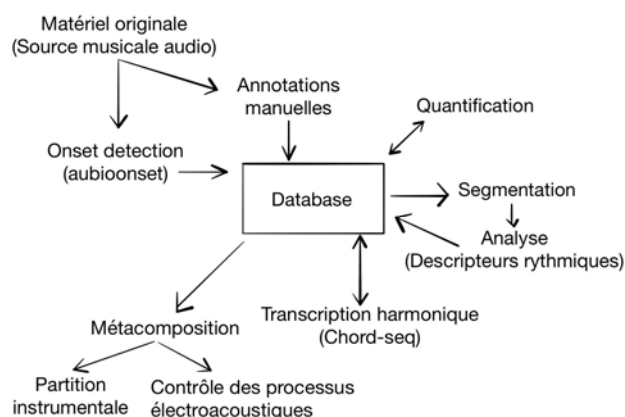


Figure 1. Schéma du flux de travaux de composition assisté par ordinateur à travers une base des données centralisée.

Le principe d'une base de données permet en outre la coordination entre typologies d'outils différents. Les résultats de chaque opération peuvent être réintégrés dans la base de données et fonctionner comme *input* pour les opérations successives.

4. L'ACTE PRÉ-COMPOSITIONNEL

L'objectif de ce flux de travaux est la composition de nouvelle musique à partir de la recombinaison, à différents échelles, d'unités sémantiques constituées par des fragments extraits (ou générés) à partir des musiques préexistantes. L'hypothèse est que ces unités, obtenues par des moyens d'analyse automatique (MIR) soient porteuses d'une identité stylistique propre, évidente ou latente. Ce contenu stylistique peut donc, dans le processus de recomposition, se combiner avec le style propre au compositeur dans une logique d'« hybridation »

Dans son traité sur le style musicale, Leonard B. Meyer [12] affirme que la perception d'un style est liée à la récurrence des formules (*patterns*) qui investissent toutes les composantes de la grammaire musicale (rythme, hauteur, timbre, etc.). Dans l'œuvre en question, l'expérience de la composition a été une sorte d'« investigation » sur ce qui peut être retenu par la perception lorsque certaines des composantes stylistiques sont dissociées, pulvérisées dans leurs unités minimales et recombinaison de façon algorithmique selon des principes qui, bien qu'ils soient tirés du matériel original, répondent à une logique d'organisation différent de celle propre au genre de départ.

Le *noise rock* se caractérise principalement par le timbre et le rythme ; pour cette raison ces deux éléments sont cruciaux dans le processus de recomposition. La présente description se focalise sur la dimension du rythme car plus facile à représenter de façon symbolique.

4.1. Le matériel de départ et les annotations manuelles

La partie instrumentale de « Confused Idols & Sexy Killers » est écrite à partir des matériaux extraits de la première piste intitulée « She's in à Bad Mood » de l'album « Sonic Youth – Confusion Is Sex (Plus Kill Yr. Idols) », réédition du 1995 des deux EP publiées séparément en 1983 par Geffen Records, code de catalogue GED 24511.

La piste audio est préliminairement divisée en 13 sections correspondant aux différentes unités du discours musical (fig. 2). Ensuite, les unités métriques de chaque section sont annotées manuellement (*beats* et *measure*) et stockées dans la base des données (table *measure_markers*) pour être récupérées lors de la phase de quantification rythmique.

Section:	01 Intro A1	01b Intro A2	02 Intro B	03 Intro C	04 Verse 01	05 Bridge 01	06 Verse 02
Mesures:	8	7	16	8	16	16	16
Start Time:	00:00:01.11	00:00:16.77	00:00:30.79	00:00:59.87	00:01:14.83	00:01:43.41	00:02:15.76
Durées (s)	15.65	14.02	29.09	14.43	29.11	32.35	28.83
BPM	122.65	119.84	132.02	133.04	131.93	118.7	133.19
Section:	07 Bridge 02	08 Bridge 02B	09 Coda A	10 Coda B1	11 Coda B2	12 Coda C	13 Final Chord
Mesures:	16	18	18	8	6	16	n.a
Start Time:	00:02:44.59	00:03:12.49	00:03:44.18	00:04:16.36	00:04:30.43	00:04:40.95	00:05:08.66
Durées (s)	27.89	31.69	32.18	14.07	10.52	27.71	n.a
BPM	137.66	136.32	134.25	136.145	136.85	138.58	n.a

Figure 2. Segmentation de la première piste : « She's In a Bad Mood ».

Pour chaque section sont ensuite rajoutées une ou plusieurs annotations manuelles qui proposent des segmentations possibles à l'échelle de micro/moyenne-forme. Ces segmentations sont les résultats d'une évaluation subjective opérée par le compositeur et correspondent à peu près aux événements temporels qui peuvent être retenus en mémoire après une première écoute. Cette simplification de la séquence des accents, enregistrée dans la base des données, est ensuite utilisée comme point de référence pour l'évaluation et la hiérarchisation des éléments extraits automatiquement.

4.2. La segmentation temporelle automatique

Une des raisons qui m'ont poussé à travailler sur des sources de Sonic Youth est la complexité de leurs textures sonores, qui sont toujours à la limite entre le son harmonique, inharmonique et le bruit pur. Le flux du discours musical est construit par la fusion d'éléments acoustiquement très différents : les impulsions de la batterie, les lignes de basse et les harmonies souvent très dissonantes des guitares électriques. Pourtant, chaque méthode de détection des attaques présente des points forts et de points faibles en fonction du matériel de départ, l'idée est donc d'éviter tout compromis dans cette phase initiale et procéder à l'analyse du matériel par tous les algorithmes à disposition. La base de données aboutit donc le rôle d'un intermédiaire centralisé qui organise les opérations successives d'évaluation et de réduction pour fournir à l'utilisateur un nombre restreint d'informations qui seront effectivement exploitables pour la composition.

Parmi les nombreuses bibliothèques de MIR qui fournissent des outils de détection des attaques (*onset detection*) le choix s'est porté sur la bibliothèque *Aubio* développée par P. Brossier [5] car tout d'abord l'outil *aubioonset* regroupe plusieurs algorithmes de détections qui sont optimisés pour différentes qualités sonores et les avantages ou désavantages en rapport aux différentes sources sont bien documentées [7]. Le logiciel opéré en *ligne de commande* par le *terminal* de MacOS, qui permet l'automatisation des processus d'analyse (*batch processing*). Il est ainsi disponible comme portage pour Max/MSP par Volker Böhm (ESB Basel). Ses paramètres principaux sont la taille de la fenêtre d'analyse (FFT) et son taux d'avancement, l'algorithme de détection (*energy*, *HFC*, *Specdiff*, *Specflux*, *Phase*, *Complex*, *Kl*, *Mkl*), le seuil de détection (en valeur normalisé entre 0. et 1.), le temps minimum entre les événements détectés et le niveau considéré comme silence.

La taille de FFT et d'avancement ont été fixées car elles affectent simplement la résolution fréquentielle et temporelle de l'analyse. Puisqu'il s'agit d'une analyse du rythme, la précision du temps est considérée comme prioritaire sur celle des fréquences. Le niveau de silence est aussi fixé, à -40dB pour éviter des détections au début et à la fin de chaque piste. Ainsi le temps minimum (fixé à 20 ms) donne une très bonne précision temporelle, mais évite de doubles détections du même événement (surtout dans le cas des rebondissements de caisse claire ou des vibratos de guitare). Les paramètres qui ont été combinés sont donc les 8 méthodes et 21 niveaux de sensibilité entre le maximum de 1 et le minimum de 0.1 par une taille de saut de 0.05. Pour chaque section nous avons donc 161 analyses effectuées.

La fonction `aubioonset` produit des listes de valeurs qui correspondent aux positions temporelles des événements détectés. Ces données numériques sont transcrites sous forme de notation spatiale proportionnelle `bach.roll` (fig. 3).

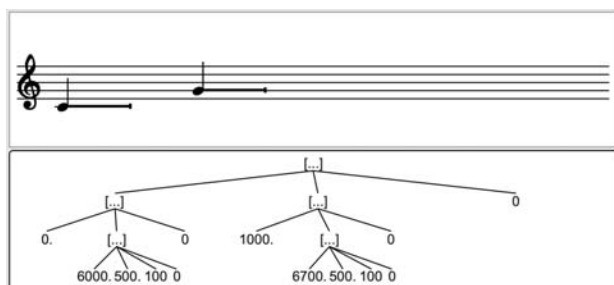


Figure 3. Représentation d'une séquence `bach.roll` sous forme de notation musicale et de liste « LISP like ».

Chaque transcription est ensuite enregistrée dans la table `aubioonset_rolls` avec d'autres données qui permettent l'organisation et la récupération au moyen des queries SQL :

- Numéro de section dans la pièce originale [`section`]
- Méthode d'analyse employée [`method`]
- Sensibilité de l'analyse [`threshold`]
- Nombre d'événements détectés [`events`]
- Étiquette libre [`tag`]

```
[ aubioonset_roll_id 1 ]
[ section 1 ]
[ method complex ]
[ threshold 0.01 ]
[ events 67 ]
[ tag BasicSet ]
[ body
  [
    [ 0.
      [ 6000. 932. 100 0 ]
      0 ]
    [ 932.
      [ 6000. 847. 100 0 ]
      0 ]
    [ 1779.
      [ 6000. 92. 100 0 ]
    ]
  ]
]
```

Figure 4. Exemple d'une transcription enregistrée dans la table `aubioonset_rolls`.

Une fois que les données « brutes » sont stockées sous une forme musicalement reconnaissable, le compositeur peut procéder à une évaluation préliminaire de la saillance musicale des résultats obtenus.

4.3. Évaluation préliminaire

Un premier filtrage est opéré en fonction du nombre d'événements détectés. Certaines séquences obtenues avec une souille ou *threshold* très haute (proche de 1.0) contiennent une quantité d'événements assez limitée, mais peuvent parfois afficher des segmentations intéressantes au niveau de la forme moyenne. Celles au-dessous de 6-8 événements par section sont éliminées car trop génériques pour fournir des morphologies vraiment intéressantes sous le profil du rythme. Il en va de même discours pour les séquences obtenues par un *threshold* très bas (proche de 0.01) qui génèrent des séquences trop chaotiques. Pour ce travail de pré-composition j'ai décidé de garder de manière empirique que les séquences entre 8 et 150 événements.

Successivement, les séquences sont confrontées entre eux par rapport aux paramètres qui les ont générées. Cette évaluation permet au compositeur de s'emparer des paramètres d'analyse à travers une approche empirique et « synthétique », fondé sur la corrélation entre paramètres et caractéristiques musicaux qui en dérivent.



Figure 5. Représentation par `dada.cartesian` de l'ensemble des analyses pour la section 3.

La visualisation d'une base de données dans une interface graphique multidimensionnelle (position sur les axes horizontale et verticale, dimension, couleur et forme) permet l'évaluation visuelle de la relation entre les deux paramètres principaux (*method* et *threshold*) et la quantité d'événements détectés, afin de déterminer leur influence sur la densité du matériel musical obtenu.

nu (fig.5). Dans l'exemple la clause `WHERE=3` limite la visualisation aux séquences extraites par la troisième section de la pièce originale. L'axe horizontal représente les différentes méthodes (dans l'ordre : *complex*, *energy*, *hfc*, *kl*, *mkl*, *phase*, *specdiff*, *specflux*) et l'axe vertical le nombre d'évènements détectés. Dans cette disposition on peut observer la distribution des évènements en relation à la progression linéaire du *threshold*, qui relève donc le comportement des méthodes d'analyse en rapport au matériel fourni.

La quantité des évènements offre une estimation générale de la densité des séquences, mais ne donne aucune indication sur la qualité des rythmes. Des informations plus intéressantes d'un point de vue musicale peuvent être obtenues grâce à l'analyse paradigmatique des différentes transcriptions pour chaque section.

À l'aide du langage de *query SQL* nous pouvons sélectionner des groupes de séquences qui présentent des caractéristiques similaires :

Dans cet exemple la séquence avec le plus grand nombre d'évènements dans l'intervalle spécifié (entre 70 et 90) est récupérée par chaque méthode et superposée aux autres et à la séquence de contrôle transcrite manuellement (fig. 6).

```
query SELECT * FROM aubiconset_rolls WHERE section = 3 AND
tag = \"BasicSet\" AND events BETWEEN 70 and 90 group by
\"method\" order by \"method\"
```



Figure 6. Disposition paradigmatique des transcriptions de la même section et avec densités approximativement égales obtenues par des méthodes différentes. La séquence de contrôle annoté manuellement est la plus en bas.

Suivant le même principe, d'autres entités peuvent être mises en relation comme les rythmes de plusieurs sections ou générées par des seuils de sensibilité différents (fig. 7 et 8).



Figure 7. Disposition paradigmatique de 3 séquences obtenues par la même section et par la même méthode mais avec sensibilités différentes.

À partir de cette organisation paradigmatique il est possible de dériver d'autres structures rythmiques « secondaires » par intersection (en retenant tout ce qui est

commun à un set de transcriptions) ou par soustraction (ce qui n'est pas présent dans une séquence et pas dans les autres). Les différentes combinaisons produisent des polyphonies rythmiques qui constituent la plupart du matériel précompositionnel de la partie électronique.



Figure 8. Labélisation des évènements commun à deux séquences.

4.4. Quantification rythmique

La librairie *bach* inclut un objet *bach.score* qui implémente une représentation de type métrique mesuré. La syntaxe de *bach.score* est similaire à celle de *bach.roll*, mais substitue aux valeurs d'*onset* et *dérations* en millisecondes un *arbre rythmique* exprimé comme proportion à travers des nombres rationnels (fig. 11).

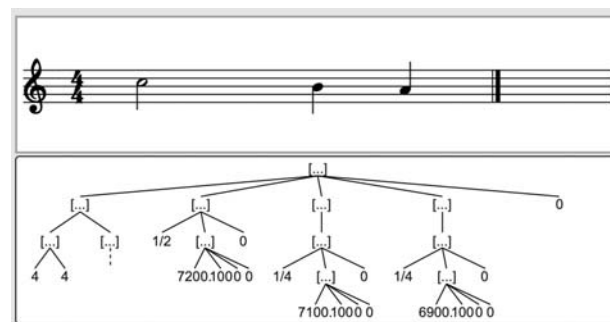


Figure 9. Représentation d'une séquence *bach.score* sous forme de notation musicale et de liste « LISP like ».

Par la fonction *bach.quantize* il est donc possible d'opérer une quantification rythmique et une transcription en notation traditionnelle. Ce processus est en grande partie automatisé et se réalise grâce au principe des « boîtes rythmiques » (*rhythmic boxes*) [2], mais tient également compte des indications fournies par l'utilisateur sous forme de marqueurs de mesure et sous-division.

4.5. Segmentation

L'annotation manuelle du rythme est donc employée comme séquence de « contrôle » auditif et visuel et pour fournir à l'algorithme des repères sous forme de marqueurs de mesure (fig. 10).





Figure 10. Groupe des séquences dans la forme originale et après la quantification rythmique. Les marqueurs en haut deviennent barres de mesure.

La collecte du matériel précompositionnel ne se limite pas aux séquences dans leur entier mais demande une investigation plus profonde, à l'échelle de la micro-temporalité, pour pouvoir successivement remodeler ces matériels de façon plus libre. Cette investigation commence par l'opération de découpage des séquences qui correspondent aux sections de la pièce originale (8 ou 16 mesures principalement) en unités plus petites.

La librairie *dada* contient la fonction *dada.segment* qui permet la segmentation d'un flux musicale (*score* ou *roll*) en sous-unités. L'objet permet une segmentation en unités fixes ou variables (à travers des *labels* ou *markers*) que l'utilisateur peut définir de façon algorithmique.

Plusieurs solutions ont été testées pour obtenir une segmentation significative sur le plan perceptuel et le choix finale s'est porté sur un algorithme récursif de *distance clustering*. Cette méthode est conceptuellement simple et facile à implémenter dans un langage de programmation graphique comme Max/MSP + bach (fig. 11).

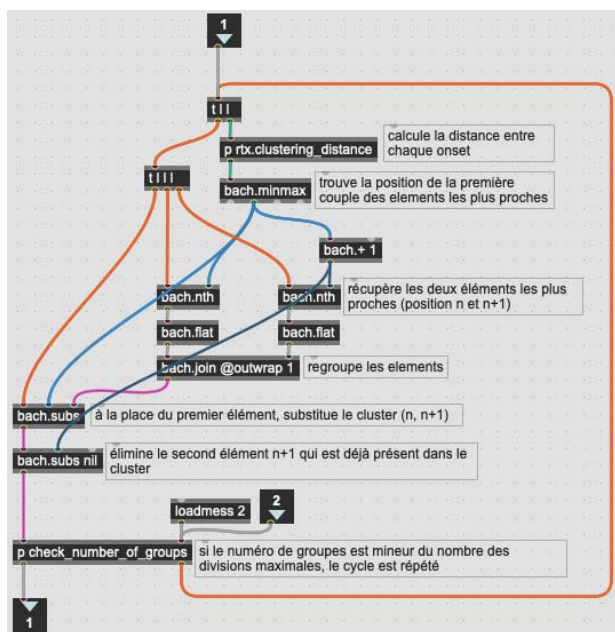


Figure 11. Implémentation en MAX/MSP et librairie bach de l'opération de clustering des deux événements plus proches.

Dans sa forme simple (non récursive) il s'agit de calculer un vecteur des distances, trouver les deux éléments le plus proches et les regrouper dans une liste. Ensuite les deux éléments sont éliminés et substitués par le

groupe et l'opération est répétée jusqu'au nombre de groupes souhaité. Dans le cas présent, étant donné qu'il s'agit d'une distribution sur une dimension (« ligne » du temps) il suffit de calculer la distance entre chaque onset et le précédent en tenant compte uniquement du premier et dernier élément de chaque regroupement.

L'agglomération « simple » est très efficace pour une segmentation de l'espace, mais ne l'est pas pour une segmentation du temps musicale. Tout musicien fait l'expérience d'une perception du rythme qui n'opère pas de façon linéaire, mais selon un modèle « à arbre », c'est à dire par divisions et sous-divisions des unités.

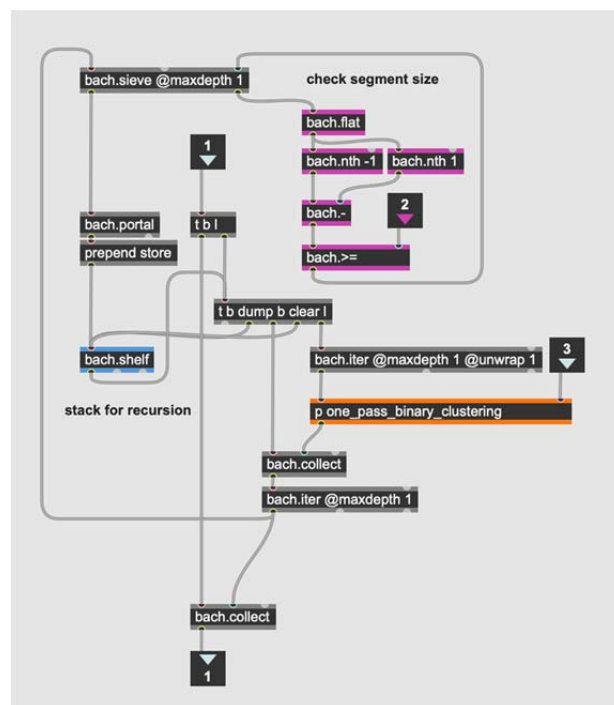


Figure 12. Implémentation en MAX/MSP et librairie bach de l'algorithme de clustering récursif. L'objet *one_pass_binary_clustering* correspond au code de fig. 11.

Par conséquent, nous pouvons obtenir une segmentation plus intéressante sur le plan musical grâce à une application récursive du même principe : l'utilisateur définit le nombre de segmentations pour chaque itération (généralement une division binaire ou ternaire) et chaque segment obtenu à partir de la séquence dans son entier est à nouveau divisé jusqu'à atteindre une taille minimale qui peut être spécifiée en millisecondes ou en nombre d'évènements par segment. De cette façon, les irrégularités de distributions au niveau générale ne sont pas prises en considération pour la segmentation au niveau local et le résultat est donc très proche des segmentations à l'écoute (fig.13).



Figure 13. Résultat de la segmentation automatique d'une séquence.

De plus, les séquences quantifiées en notation métrique (*scores*) subissent un autre type de segmentation, car il est possible d'introduire dans le processus la notion de *beat* et de mesure qui peut générer une plus grande quantité de matériel « dérivée ». Les séquences originales sont coupées en unités de 1, 2 et 4 mesures et avec une taille d'avancement d'une noire. Le nombre des fragments obtenus est successivement filtré par des contraintes, qui éliminent les fragments qui commencent par une pause, qui ont moins d'un certain nombre de notes par mesure ou qui sont trop similaires à ces qui sont déjà présents dans la base de données.

4.6. Analyse des caractéristiques et descripteurs rythmiques

Lors de la phase de segmentation, l'objet *da-da.segment* permet également de labelliser chaque fragment avec des données spécifiées par l'utilisateur ou obtenues par analyse du contenu, à travers des algorithmes implémentés sous forme de *patch* Max/MSP.

Avant d'être stocké dans la base de données, chaque segment est donc analysé et étiqueté selon des caractéristiques qui permettent la récupération des motifs à partir de leurs attributs.

Le choix des caractéristiques (*features*) à enregistrer dans la base de données est un acte précompositionnel au sens fort du terme, car il définit les axes d'organisation de la phase d'écriture : ces caractéristiques deviennent, dans l'acte de composition, les paramètres des processus de structuration morphologique et les gradients d'énergie rythmique des différents « moments » et le déroulement de la forme globale dépendent des paramètres déterminés à cette étape du travail et représentés sous forme de colonnes dans la base de données (fig. 14).

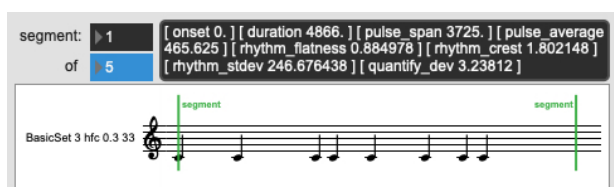


Figure 14. Résultat de l'analyse automatique d'une séquence.

Certaines caractéristiques simples fournissent des indications de bas niveau à partir de la distance entre les événements (*inter-onset time*) :

- *Fragment_onset* : position de la séquence dans l'enregistrement original
- *Pulse_span* : distance entre le premier et dernier événement de la séquence.
- *Pulse_min_dur* : distance minimale
- *Pulse_max_dur* : distance maximale
- *Pulse_average* : distance moyenne
- *Pulse_median* : médiane des distances
- *Pulse_stdev* : déviation standard des distances

Pour pouvoir imaginer des processus musicaux de façon plus intuitive, des descripteurs rythmiques de plus

haut niveau sont également proposés. Deux descripteurs de niveau « moyen » sont introduits à partir de leur équivalent spectral, communément utilisés dans l'analyse du son :

- Le *rhythm_crest* est une sorte de *crest factor* calculé à partir de la distance entre les événements rythmiques et défini selon la formule classique qui s'applique au signal électrique comme rapport entre la valeur absolue la plus élevée (*peak*) et la moyenne des valeurs :

$$C = \frac{|x_{\text{peak}}|}{x_{\text{rms}}} = \frac{\|x\|_{\infty}}{\|x\|_2}$$

- Le *rhythm_flatness* définit pour le signal audio la platitude du spectre. Il est ici calculé comme rapport entre moyenne géométrique et la moyenne arithmétique de la distance entre les événements :

$$\text{Flatness} = \frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\frac{\sum_{n=0}^{N-1} x(n)}{N}} = \frac{\exp\left(\frac{1}{N} \sum_{n=0}^{N-1} \ln x(n)\right)}{\frac{1}{N} \sum_{n=0}^{N-1} x(n)}$$

Ces descripteurs nous informent du type d'irrégularité rythmique des séquences. Alors que la simple moyenne des durées peut être équivalente pour des rythmes qui ont une irrégularité distribuée sur toute la durée ou présentent des zones régulières et d'autres fortement irrégulières, les notions de « crête » (*crest*) et de planéité (*flatness*) nous permettent de discriminer ces deux cas avec plus de précision.

Cependant, pour les séquences non quantifiées (*rolls*), ces notions, calculées simplement à partir des variations de durée des événements, ne donnent aucune information sur la perception du type d'irrégularité relevée.

Avec les descripteurs statistiques énumérés jusqu'ici, une séquence perçue comme aléatoire et une autre, articulée dans un rythme très complexe, peuvent afficher des résultats similaires. En revanche, du point de vue musical, la différence est considérable.

Ici, je propose un descripteur qui exprime, à travers une valeur numérique, dans quelle mesure les variations des durées dans la séquence sont rapportables à une pulsation de base. En pratique, cela équivaut à indiquer dans quelle mesure la séquence ressemble à son équivalent quantifié. Ce descripteur, nommé *quant_likely* est calculé à partir d'un principe qui montre certaines similarités avec la méthode de quantification proposée par Agon et Asssayag dans la librairie *Kant* pour OpenMusic [1] (Fig. 15). Les durées sont ordonnées par ordre de valeurs croissant, puis le rapport entre chaque durée et les durées inférieures est calculé. Nous obtenons donc un vecteur qui exprime combien de fois chaque durée est contenue dans les durées supérieures. Pour chaque valeur, une approximation à la fraction de dénominateur 2 et 3 est calculée et l'erreur d'arrondi la plus petite est retenue. Ensuite la variance du vecteur des erreurs est calculée, ce qui nous donne la mesure dans laquelle les rapports de durées s'approchent aux multiples ou sous-

multiples d'une division binaire ou ternaire des valeurs. Grâce à ce principe, il est possible d'estimer, du point de vue de la perception, si une séquence est plus ou moins strictement « à tempo ».

Une version plus « stricte » peut être obtenue en utilisant une simple troncature au seul dénominateur 2. Dans ce cas, le système tend à considérer les triplets comme non quantifiés. Autres versions plus « tolérantes » peuvent inclure les divisions par 5, 6 ou 7.

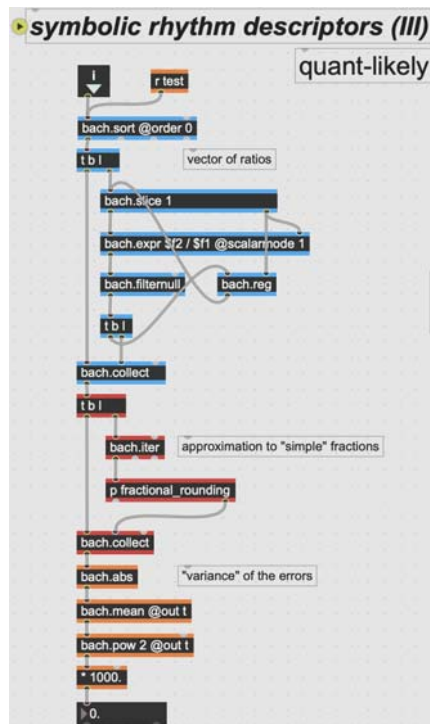


Figure 15. Algorithme *quant_likely* sous forme de code MAX/Msp.

Un simple système de *benchmarking* a été mis en place pour mesurer l'efficacité du descripteur. Des séquences aléatoires sont générées puis quantifiées avec des valeurs plutôt fines (inclusion des divisions par 5, 6 et 7). La version originale et la version quantifiée sont analysées par le descripteur et le résultat montre une forte cohérence : la version quantifiée produit une valeur de *quant_likely* 2 à 3 fois plus élevée (fig 16).

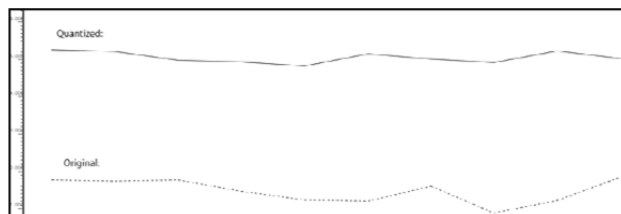


Figure 16. Valeur de *quant_likely* pour les séquences aléatoires avant (ligne pointillée) et après (ligne pleine) la quantification.

5. (MÉTA)COMPOSITION

Une large partie de la partition instrumentale et du support électronique de « Confused Idols and Sexy Killers » est donc réalisé à partir de la recombinaison diachroniques et synchroniques du corpus rythmique organisé par la base de données. Les processus de CAO sont, dans ma démarche créatrice, un support pour les choix compositionnels sans en être un substitut.

J'ai donc souvent utilisé la base de données comme un vaste catalogue de motifs et de fragments pour guider l'écriture manuelle de façon intuitive et contrôlée par le *feedback* de l'écoute « intérieure ».

Dans de nombreuses autres sections de la pièce, au contraire, j'ai opté pour une automatisation du processus d'assemblage des motifs à travers des techniques de méta-composition offerts par l'objet *cage.meta*, inclus dans la librairie *cage* pour Max/MSP [9].

Cet objet permet la construction de méta-partitions qui associent à chaque événement (note) un ou plusieurs processus génératifs implémentés sous forme de code visuel (*patch*).

Lors de l'exécution en temps réel ou du *rendering* en temps différé, le système exécute les portions de code qui sont associées à chaque événement de la méta-partition dans sa temporalité et produit un résultat sous forme de notation traditionnelle. Si le code associé est de nature déterministe, la méta-partition représente simplement un instrument puissant pour la structuration hiérarchique de différents niveaux formels. Au contraire, si nous introduisons des éléments de variabilité, nous pouvons obtenir des partitions qui incluent une composante aléatoire contrôlée ou une composante interactive.

Lors de l'exécution en temps réel ou du *rendering* en temps différé, le système exécute les portions de code qui sont associées à chaque événement de la méta-partition dans sa temporalité et produit un résultat sous forme de notation traditionnelle. Si le code associé est de nature déterministe, la méta-partition représente simplement un instrument puissant pour la structuration hiérarchique de différents niveaux formels. Au contraire, si nous introduisons des éléments de variabilité, nous pouvons obtenir des partitions qui incluent une composante aléatoire contrôlée ou une composante interactive.

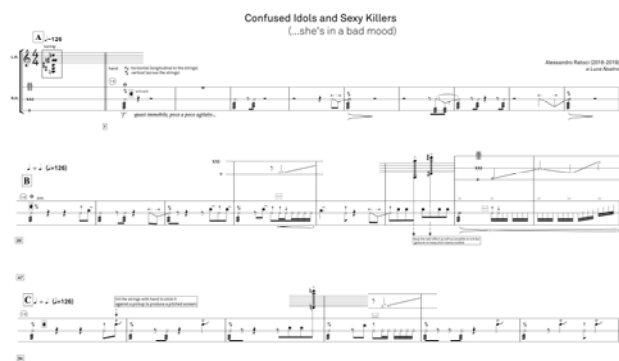


Figure 17. Comparaison du début des trois premières phrases (ou arcs) de la pièce « Confused Idols & Sexy Killers ».

Les agents ou « machines » de la méta-composition utilisées dans ce cas spécifique ont pour base l'interface *dada.cartesian*, 4.3 qui récupère les fragments dans la base de données en accord avec l'ensemble des caractéristiques demandées. L'objet permet aussi une recherche par la méthode des « *k* plus proches voisins » (*k-nearest-neighbor* ou *knn*) qui fournit un nombre de solutions proches de la combinaison de paramètres spécifiés. Le compositeur peut donc définir une trajectoire dans l'espace paramétrique défini par les colonnes de la base de données et l'outil réalise cette trajectoire à l'aide des fragments contenus dans la base de données. Cette approche permet une certaine indépendance entre matériel et processus, qui peuvent être réutilisés et combinés de

manière différente afin d'obtenir une plus grande diversité de résultats. Deux utilisations de ce principe, extraites de la première section de la pièce sont ici présentes comme illustration de ce propos.

Le premier se situe au début (lettres A-G, du début à 5'30''). La partie de guitare électrique est structurée dans une série de phrases, ou « arcs » de durée décroissante générées par assemblage des rythmes quantifiés (fig. 17). Chacun des 7 « arcs » utilise le matériel d'une seule section dans l'ordre original, comme une sorte de « réminiscence » de la *song* de Sonic Youth dans sa forme inaltérée.

La section suivante (à partir de la lettre H) propose au contraire l'accumulation de ces matériaux (section 1, 1+2, 1+2+3, etc.) qui densifie la texture sonore et rend moins reconnaissable les matériaux. En plus de la position dans le fichier original (*fragment_onset*), une autre caractéristique est employée pour piloter la direction du discours musicale : la densité moyenne des événements (*pulse_average*) augmente progressivement au sein de chaque phrase et tout au long de la section (dans un profil schématique qui ressemble à la superposition entre une itération en « dents de scie » et une croissance linéaire).

À partir de la métrique de la partie instrumentale, un second processus apparaît dans l'électronique, qui accompagne les gestes du guitariste avec des « nuages » de micro-événements réalisés par échantillonnage et synthèse granulaire. Le séquençage et la synthèse sont contrôlés par des événements discrets (MIDI notes) générés à partir de séquences non-quantifiées (*rolls*).

La combinaison entre séquences quantifiées et non-quantifiées satisfait l'exigence d'équilibre entre prévisibilité et richesse : la partie instrumentale est toujours écrite à partir des séquences métriques, qui peuvent être réalisées avec précision par l'instrumentiste ; l'interprète ajoutera toujours les nuances nécessaires à la vie et à la richesse de matière musicale. La partie électronique, qui est plus « rigide » par nature, emploie au contraire les séquences non-quantifiées qui préservent tous les nuances de l'exécution originale. Il s'agit d'optimiser la conservation des caractéristiques « analogiques » de l'interprétation originale de Sonic Youth, y compris leur *feel* et leur *groove*. Nous pouvons, d'une certaine manière, parler de « rythme concret » dans la même acception du terme « son concret » (Schaeffer 1966) [14], pris dans l'ensemble de ses caractéristiques qui le rendent unique.

Une couche de complexité s'ajoute à cette relation déjà riche entre instrument et électronique. Étant donné la quantité considérable de matériel accumulé et l'approche paramétrique de la forme, il est possible de générer une électronique qui utilise, à chaque exécution, des matériels différents mais avec des caractéristiques similaires, selon des processus stochastiques qui donnent un résultat légèrement différent, « vivant », à chaque performance.

Pourtant un dispositif de méta-composition a été implémenté dans l'environnement du concert sous forme de *device* Max For Live.



Figure 18. Interface de métacomposition par les libraires *cage* et *bach*.

Dans son fonctionnement schématique des séquences quantifiées (1) sont utilisées comme point de départ pour générer une méta-partition (2) qui contient les paramètres (*pulse_span*, *quant_likely*) pour la récupération *ad hoc* des séquences « fluides » qui sont assemblées (3) pour produire le flux MIDI.

Le son est finalement généré par un dispositif de synthèse granulaire implémenté en Max For Live et contrôlé par notes MIDI (fig. 19).

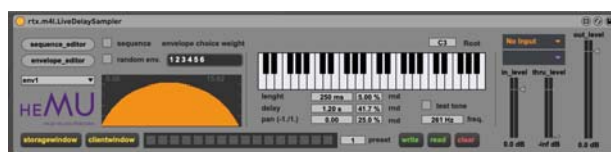


Figure 19. Dispositif Max For Live pour la resynthèse granulaire en temps réel contrôlée par notes MIDI.

6. CONCLUSION

Le travail préliminaire pour la réalisation de « Confused Idols & Sexy Killers » m'a permis de développer une série d'outils qui dépassent les nécessités immédiates de la composition de l'œuvre ; les concepts que j'ai pu développer seront certainement le fondement de futurs travaux.

La possibilité de séparer au moyen d'un une approche par base de données, le corpus de matériels (*tables*) et les principes de structuration (*queries*) rend possible leur réutilisation dans d'autres contextes de composition qui peuvent mettre en jeu des sources différentes.

Le corpus est destiné à s'élargir en incluant également d'autres entités que les séquences rythmiques, comme par exemple des fragments mélodiques ou harmoniques et leurs caractéristiques structurantes. La dimension du timbre peut être elle aussi être explorée par une approche par base de données en utilisant les descripteurs spectraux. Plus généralement, la méthode n'est pas limitée au traitement des données symboliques ; la dimension du son concret peut être également associée à travers des références aux fichiers audio et la connexion avec des outils pour la transformation électroacoustique du signal, (de façon similaire à certains outils déjà présents dans le panorama du *sound design* comme le logiciel SoundShaper développé par Robert Fraser [13] ou pour le traitement du son en temps réel.

L'exploration de différentes méthodes de segmentation fondées possiblement sur l'apprentissage automatique, l'implémentation de certains modèles

d'évaluation des transcriptions automatiques [6], ou encore l'inclusion du concept de similarité [17] dans les caractéristiques rythmiques sont d'autres possibilités de développements futurs qui pourront trouver leur place dans le flux de travaux pour mes prochaines œuvres.

7. REFERENCES

- [1] Agon, C., et coll. « Kant: a Critique of Pure Quantification », ICMC, Aarhus, Danemark, 1994.
- [2] Agostini, A., Ghisi, D. « A Max Library for Musical Notation and Computer-aided Composition », *Computer Music Journal* 39/2 (2015), p. 11-27.
- [3] Bello, J. P., et coll. « A Tutorial on Onset Detection in Music Signals », *IEEE Transactions on speech and audio processing* 13/5 (2005), p. 1035-1047.
- [4] Bogaards, N., Roebel A. « An Interface for Analysis-driven Sound Processing », Audio Engineering Society Convention 119, New York, United States, 2005.
- [5] Brossier, P. M. « Fast Onset Detection using Aubio (Brossier), Mirex 2005 ».
- [6] Brossier, P. M. « The aubio library at mirex 2006 », *Synthesis* (2006).
- [7] Brossier, P. M. « Automatic Annotation of Musical Audio for Interactive Applications », Thèse de doctorat, sous la dir. de Mark Plumbley et Mark Sandler. Queen Mary, University of London, London, 2006.
- [8] Duxbury, C., Bello, J. P., Davies, M., Sandler, M. « Complex Domain Onset Detection for Musical Signals », Proceedings of the 6th International Conference on Digital Audio Effects 120, London, United Kingdom, 2003, p. 4.
- [9] Ghisi, D., Agostini A., Maestri E. « Recreating Gerard Grisey's Vortex Temporum with cage », Proceedings of the ICMC 2016. Michigan Publishing, Ann Arbor, 2003.
- [10] Ghisi, D. « Music across Music: towards a Corpus-based, Interactive Computer-aided Composition », Thèse de doctorat. Sous la dir. de C. Agon et Pi. Couprie, Université Paris 6, Paris, 2017.
- [11] Marques, J. « An Automatic Annotation System for Audio Data Containing Music », Thèse de doctorat, sous la dir. de T. A. Poggio, Massachusetts Institute of Technology, Cambridge (MA), 1999.
- [12] Meyer, L. B. *Style and music: Theory, history, and ideology*. University of Chicago Press, Chicago, 1996.
- [13] Miranda, E. *Computer sound design: synthesis techniques and programming*. Routledge, Abingdon-on-Thames, 2012.
- [14] Schaeffer, P. *Traité des objets musicaux*. Le Seuil, Paris, 2016 (rééd. de 1966).
- [15] Solomos, M. « Les écarts entre esquisses pré-compositionnelles et œuvres: les cribles de nomos alpha (xenakis) », *Les cahiers du CIREM (Centre international de recherches en Esthétique musicale)* 40-41 (1997), p. 141-155.
- [16] Stravinsky, I. *Poetics of music in the form of six lessons*. Harvard University Press, Cambridge (MA), 1947.
- [17] Toussaint, G. T. « A Comparison of Rhythmic Similarity Measures ». ISMIR 2004, Barcelona, Spain, 2004.
- [18] Xenakis, I. *Formalized music: thought and mathematics in composition*. Harmonologia Series No. 6. Pendragon Press, Stuyvesant (NY), 1992.

Texte édité par Tom Mays