

SEQUENZA

Denis Pousseur
denis.pousseur@gmail.com

Le nom « Sequenza » est un hommage à Luciano Berio

RÉSUMÉ

Sequenza ¹ est un nouveau logiciel d'aide à la composition qui arrive à sa première version publique après une huitaine d'années de recherche et développement.

Sequenza se situe quelque part à la frontière entre deux grandes familles logicielles : les séquenceurs MIDI et les éditeurs de partition. Il possède aussi un important volet d'aide à la création, s'approchant ainsi de certains outils de recherche comme, par exemple, *Open Music* ² développé à l'IRCAM.

L'objet de cet article est de présenter ce positionnement et la manière dont le polymorphisme qui en découle est mis en application dans quelques aspects concrets de l'application. L'article commence par des considérations de bas niveau touchant à certains aspects fondamentaux de la théorie musicale, puis évolue vers des questions touchant aux plus hauts niveaux de l'application, tout particulièrement à son architecture.



Figure 1. Le logo de Sequenza

1. Sequenza est une application qui tourne exclusivement sur Mac OS X, version 10.5 (Léopard) ou ultérieure.

2. Les recherches qui ont mené progressivement au développement de Sequenza ont commencé sur la plateforme *Open Music*. Ceci explique certaines convergences de conception même si, aujourd'hui, les philosophies des deux applications sont fort différentes, et de ce fait d'ailleurs assez complémentaires. Ceci explique aussi pourquoi les couches fondamentales de Sequenza sont développées en Common Lisp comme l'est *Open Music*. À noter toutefois que la couche de présentation (GUI) de Sequenza est entièrement basée sur la bibliothèque Cocoa de Mac OS X et fait donc usage de l'Objective-C. Ceci est rendu possible par le bridge entre Lisp et Objective-C développé à Cambridge par la firme Lisworks (2008) [1].

1. INTRODUCTION

Un des objectifs de Sequenza est de permettre au compositeur de travailler sur les rendus sonore et graphique d'une partition dans un même environnement tout en lui offrant de nombreux outils d'assistance à la composition.

À la différence d'*Open Music*, Sequenza n'est toutefois pas un « métalangage » de programmation. En se positionnant au carrefour de plusieurs familles logicielles, Sequenza ne peut en effet pas développer de manière exhaustive chacune des matières qu'il traite. Il jette plutôt entre elles des passerelles qui ne sont possibles que dans un environnement intégré. Ces passerelles structurent l'application qui cherche ainsi à s'approcher au plus près du « geste » compositionnel.

L'attention portée par Sequenza à la modélisation sonore accompagne une mutation qui, peu à peu, confère à l'ordinateur le rôle que les compositeurs donnaient autrefois au piano. Tout éditeur de partition propose aujourd'hui une forme de modélisation sonore, mais il le fait en s'appuyant sur des automatismes basés sur l'iconographie sur lesquels l'utilisateur a peu de contrôle. Cette approche est attractive (la partition se joue « toute seule ») mais le résultat en est généralement peu convaincant.

Sans dédaigner cette voie, Sequenza développe d'autres stratégies. L'accès aux données sonores y est tout aussi complet qu'il l'est dans un séquenceur MIDI et de nombreux mécanismes d'interaction entre l'écrit et le sonore sont proposés. Cet aspect est évoqué à la section 4.

Sequenza peut aussi se voir comme un « générateur et processeur de musique écrite ». On peut l'utiliser à la manière d'un créateur de musique électronique : travailler la matière, la générer, la traiter, l'organiser. On peut ensuite se poser des questions plus spécifiquement liées à l'écriture et à l'interprétation. Cette façon de voir un cycle de travail dans Sequenza explique l'attention portée à l'indépendance du rendu sonore et du rendu graphique, elle explique aussi l'attention portée à l'architecture et au capacités d'organisation des matériaux évoqués à la section 5.

Enfin, Sequenza aborde les questions de théorie musicale avec la volonté d'y intégrer les apports nombreux dont le XXe siècle l'a enrichie, ce qui est particulièrement illustré par les sections 2 et 3.

2. GESTION DES HAUTEURS

Sequenza s'efforce de rechercher des éléments de convergence entre les différentes pratiques compositionnelles. Pensé pour aider le compositeur d'aujourd'hui, Sequenza ne délaisse pas pour autant une gestion classique de la tonalité.

Dans ce même esprit, Sequenza tente une approche de la microtonalité qui soit compatible avec son versant tonal. À ces approches tonale et microtonale, Sequenza ajoute encore les approches spectrale et modale, le tout formant un ensemble qui se veut le plus transparent possible pour l'utilisateur.

2.1. Hauteurs microtonales

2.1.1. Principe

Sequenza gère la microtonalité sous forme de deux modes s'ajoutant au mode chromatique compatible avec la norme MIDI (MIDI, 1982) [2] : les quarts de tons et les sixièmes de tons. Le choix de ces deux modes n'est pas arbitraire, il est lié aux choix fait pour la représentation graphique des micro-intervalles. Afin de créer une progression fluide entre sa dimension tonale et microtonale, Sequenza exprime les micro-intervalles sous forme de trois pas relatifs aux hauteurs chromatiques :

(-1, 0, 1) Hauteur baissée, normale ou haussée

Chaque type d'altération possède donc deux versions graphiques alternatives à la version normale, pour pouvoir s'adapter à ces degrés intermédiaires. La valeur haussée et la valeur baissée sont représentées par une petite flèche montante ou descendante au sommet ou au bas de l'altération (ou du bécarre pour les notes non altérées). Le double dièse et le double bémol n'ont respectivement qu'une version baissée et une version haussée.

Pour l'expression textuelle de la hauteur, les valeurs haussées et baissées seront symbolisées par un signe «+» ou «-» ajouté à la suite du nom de la note. Par exemple C \sharp 3+ signifie un quart de ton ou un tiers de ton au-dessus de C \sharp 3, ceci dépendant du mode microtonal choisi.

2.1.2. Mise en pratique

Si le mode microtonal choisi est «quarts de ton», la valeur haussée et la valeur baissée du degré chromatique supérieur sont simplement en «enharmonie» (c'est-à-dire que leur rendu sonore est identique).

Si le mode microtonal est «sixièmes de ton», la valeur haussée représente le premier tiers de demi-ton et la valeur baissée du degré chromatique supérieur représente le second tiers de demi-ton.

De cette manière, Sequenza est capable de mélanger la logique d'écriture tonale avec l'apport de la microtonalité. Par exemple, un quart de ton au-dessus de do peut se représenter au choix par :

si \sharp +, do+, do \sharp -, réb-, ou même ré \flat +

2.1.3. Rendu sonore de la microtonalité

Les ingénieurs d'Apple ont prévu dans le protocole de communication avec le «synthétiseur DLS» de la bibliothèque «Core Audio» un format MIDI «étendu» (Apple Inc. 2001) [3] qui permet l'utilisation de valeurs fractionnelles en lieu et place des valeurs entières prévues par la norme MIDI. La hauteur reste codée comme un chiffre allant de 0 à 127 mais, grâce à la partie fractionnelle, on peut exprimer les valeurs microtonales en allant d'ailleurs bien au-delà de la définition offerte actuellement par Sequenza. Sequenza utilise cette fonctionnalité pour le rendu sonore de la microtonalité. Quand on travaille avec des instruments virtuels d'autres constructeurs, Sequenza représente les hauteurs microtonales, mais elles ne peuvent pas être entendues en raison des limitations du protocole MIDI.

L'ajout d'un système plus universel, par exemple inspiré de celui développé par OM (OM 6.5, Microintervals) [4], reste une solution possible pour le futur.

On peut aussi rêver d'un protocole MIDI «étendu» qui se généralise (HD-MIDI, 2008) [5], ou d'une reconnaissance plus répandue de l'extension à la norme prévue dès 1995 par la MMA (MIDI Tuning Messages, 1995) [6] afin de permettre la représentation d'échelles de hauteur non dérivées du système tempéré occidental.

2.1.4. Développements possibles

La limite de définition microtonale est fixée pour des raisons liées à l'expression graphique des altérations. Mais une version mieux définie du même système pourrait être envisagée en ajoutant les concepts de «demi haussé» et «demi baissé». Ceci enrichirait l'offre des modes existants aux huitièmes et dixièmes de ton sans briser pour autant la compatibilité tonale.

2.2. Hauteurs tonales

La hauteur «tonale» (j'entends pas là, une hauteur qui - à la différence de la hauteur MIDI - inclut son expression diatonique et son altération) est indispensable à la représentation graphique de la partition. Pour stocker cette hauteur, Sequenza utilise un système de codage qui lui permet de la manipuler ensuite avec simplicité.

Dans Sequenza, ce que l'on appelle généralement «transposition tonale» (Letz, Fober, Orlarey, 1999) [7] est nommé «transposition modale» (un concept évoqué au point 2.4), alors que la transposition tonale recouvre un concept différent : c'est une «transposition chromatique» qui présente en plus l'avantage de préserver le dessin diatonique et la logique des altérations, c'est-à-dire la logique d'écriture. Concrètement, la hauteur de la note est codée, hors octave, dans une plage allant de 2 à 32 et représentant 3 cycles des quintes (incomplets) en même temps que 5 cycles diatoniques (incomplets). Fa est 14, do est 15, sol est 16... fa \sharp est 21, fa $\sharp\sharp$ est 28, etc. (voir figure 2). Plus on monte plus on va vers des altérations «haussées», et

inversement. C'est un système assez proche de ceux évoqués par Rasch (2000) [8].

0	(F $\flat\flat$)	(C $\flat\flat$)	G $\flat\flat$	D $\flat\flat$	A $\flat\flat$	E $\flat\flat$	B $\flat\flat$
7	F \flat	C \flat	G \flat	D \flat	A \flat	E \flat	B \flat
14	F	C	G	D	A	E	B
21	F \sharp	C \sharp	G \sharp	D \sharp	A \sharp	E \sharp	B \sharp
28	F $\sharp\sharp$	C $\sharp\sharp$	G $\sharp\sharp$	D $\sharp\sharp$	A $\sharp\sharp$	(E $\sharp\sharp$)	(B $\sharp\sharp$)

Figure 2. Tableau de codage des altérations

Dans l'espace décrit par la figure 2, les manipulations se font par de simples additions et soustractions d'index. Les notes entre parenthèses sont hors des limites acceptées. Hausser ou baisser une altération d'un degré revient à ajouter ou à soustraire 7 (déplacement vertical dans le tableau). Trouver l'enharmoine d'une note revient à ajouter ou à soustraire 12 ou 24 au code.

Tant que l'opération ne mène pas le code en dehors de la plage [2, 32], elle est « possible » (au sens de la notation conventionnelle). Sinon, elle n'est possible qu'en remplaçant une hauteur par son enharmonie.

Par exemple, si une valeur devient 34 après manipulation (Si $\sharp\sharp$ – une valeur non représentable car $34 > 32$), il faut la ramener dans la plage valide en lui soustrayant 12. Elle devient donc 22 c'est-à-dire do \sharp , l'enharmoine la plus proche.

Du point de vue de l'utilisateur, la transposition est présentée de manière conventionnelle (ajout d'une seconde diminuée ou d'une quinte augmentée par exemple). Quand Sequenza réalise une « transposition tonale » explicitement et qu'une impossibilité est rencontrée, l'utilisateur a le choix entre abandonner la procédure ou accepter les enharmonies nécessaires au calcul.

Finalement, après traitement, on peut déduire des codes résultants les notes diatoniques (hors octave) et les altérations, c'est-à-dire les attributs graphiques qui permettent de représenter la hauteur.

2.3. Hauteurs spectrales et fréquence

À divers endroits de l'application, les hauteurs sont aussi exprimées en hertz. La hauteur en hertz n'est pas stockée, elle est simplement déduite de la hauteur de la note (sur base d'un la à 440 Hz). De ce fait, on ne peut pas fixer la hauteur d'une note sur n'importe quelle valeur en hertz, on ne peut la fixer que sur une fréquence correspondant à une hauteur musicale. L'approximation dépendra du mode microtonal choisi. Pour des calculs spectraux, l'utilisation du mode microtonal le mieux défini est donc souhaitable. Divers outils de traitement ou de génération permettent aussi de manier la hauteur sous forme d'une valeur harmonique se référant à une fondamentale. Il est ainsi possible de faire de la « transposition spectrale » (addition d'un certain nombre de degrés harmoniques aux hauteurs existantes), ou de générer des dessins mélodiques sous forme de suites d'harmoniques. Ces calculs sont relatifs à la fondamentale choisie par l'utilisateur, un choix

qui sera évidemment déterminant pour le résultat.

2.4. Hauteurs modales et échelles

Une échelle est un « choix de sons disposés par fréquence croissante ou décroissante. À la différence du mode, l'échelle n'établit aucune hiérarchie entre les sons » (Hakim et Dufourcet 1995) [9]. L'échelle est donc une couche d'abstraction supplémentaire à la notion de « mode ». Sequenza propose un outil très complet qui permet de créer et de manipuler des échelles « hors temps ». Les échelles sont présentées comme allant du plus grave au plus aigu mais elles ne doivent pas nécessairement être constituées d'une suite de hauteurs triées : le dessin peut être constitué de segments montants et descendants, par exemple à la manière de l'« extrapolation » ou de l'« infrapolation » décrites par Slonimsky (1947) [10]. Ce type d'échelle n'est pas conseillé pour un traitement faisant appel à l'approximation, mais peut être intéressant quand le traitement visé travaille sur base de l'indexation, c'est le cas de la « projection modale » (projection des index d'un dessin mélodique d'une échelle dans une autre) ou de la « transposition modale » (addition d'index dans une échelle donnée).

2.4.1. Modes

Un mode est une suite d'intervalles (au minimum un) qui se répète en se transposant selon la somme de ses intervalles, généralement l'octave mais pas nécessairement, et qui dispose d'un degré fondamental. Les échelles qui répondent à cette définition, par exemple le mode diatonique, sont développées dans Sequenza comme couvrant tout l'ambitus de l'instrument.

Si l'intervalle entre deux degrés fondamentaux du mode est autre que l'octave, on parle alors de mode « non-octaviant ». De nombreux modes de ce type ont été décrits par Slonimsky (1947) [10] et sont très simplement réalisables dans Sequenza.

2.4.2. Échelles fixes

Dans Sequenza, une « échelle fixe » est constituée d'une suite de notes indéterminées, possiblement sans aucun principe de répétition. À la différence du mode, elle ne se répète pas et ne couvre donc pas nécessairement tout l'ambitus de l'instrument.

On peut utiliser les échelles fixes à des fins compositionnelles, mais aussi comme aide à l'introduction de données, par exemple pour ne laisser écrire dans un jeu de portées que les hauteurs incluses dans un set de percussions donné (voir 2.4.5).

2.4.3. Échelle des harmoniques

Dans Sequenza, l'échelle des harmoniques est représentée sous forme d'une échelle fixe en sixièmes de ton montant jusqu'à l'harmonique 64.

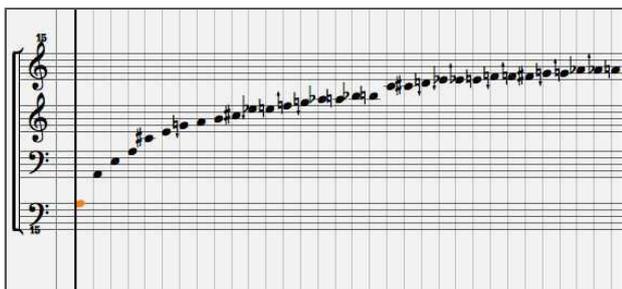


Figure 3. Vue partielle de l'échelle des harmoniques dans l'éditeur d'échelle

2.4.4. Stockage, parenté et identité

Tous ces types d'échelles sont représentables dans Sequenza et peuvent être sauvegardés sur le disque dur sous forme de bibliothèques indépendantes de la session. Chaque échelle possède son propre mode microtonal, elle impose aussi à la partition sa propre manière de représenter les altérations, ceci pouvant être ensuite adapté note à note par l'utilisateur. Chaque instrument possède une échelle (par défaut l'échelle diatonique) qui détermine la notation des altérations. Les séquences mono-instrumentales peuvent également contenir une ou plusieurs échelles réparties dans le temps, ce qui permet, notamment, à l'application de gérer les changements d'armure. Quand une échelle est attachée à la séquence, elle a priorité sur celle de l'instrument parent. Il est possible de transposer une échelle, ou de modifier sa fondamentale, ainsi pour exprimer les 7 modes diatoniques dans les 12 tonalités chromatiques, on n'a en fait besoin que d'une échelle dont on adaptera deux paramètres : la transposition et le degré fondamental.

2.4.5. Usage

De même que l'application utilise le magnétisme rythmique pour aider à l'introduction des rythmes (voir point 3.2), les échelles de Sequenza lui permettent d'opérer un magnétisme « harmonique », évoluant potentiellement dans le temps, qui peut être plus ou moins contraignant. Il peut ne servir qu'à aider l'introduction de notes dans la partition, ou à leur transposition interactive, mais il peut aussi magnétiser de manière systématique interdisant de fait l'existence de toute note n'appartenant pas à l'échelle dans la séquence. Ce magnétisme étant « non-destructif », il permet de tester l'alignement avec des échelles différentes : le calcul repart toujours des hauteurs originales qui sont conservées intactes - et qui peuvent d'ailleurs être rétablies à tout moment en désactivant simplement le magnétisme. Les échelles peuvent aussi servir à la sélection (ne sélectionner que les notes appartenant à une échelle donnée par exemple), au traitement ou encore à la génération de matériaux musicaux. Dans ce cas, les hauteurs, dont on a déjà vu qu'elles pouvaient être exprimées comme des notes ou des harmoniques, sont exprimées comme les degrés (les index) d'une échelle. Le script de génération produira dans ce cas un résultat dépendant

du contexte, c'est-à-dire de l'échelle courante.

3. GESTION DU TEMPS

3.1. Mode proportionnel

Conservant là une filiation avec *Open Music* (OM 6.5 Score Objects) [11], Sequenza s'organise essentiellement en deux modes de représentation de la partition, l'un plutôt destinée au travail sur la modélisation sonore et sur la composition et l'autre sur le rendu graphique et la mise en page. Dans le mode « proportionnel » (appelé « notation

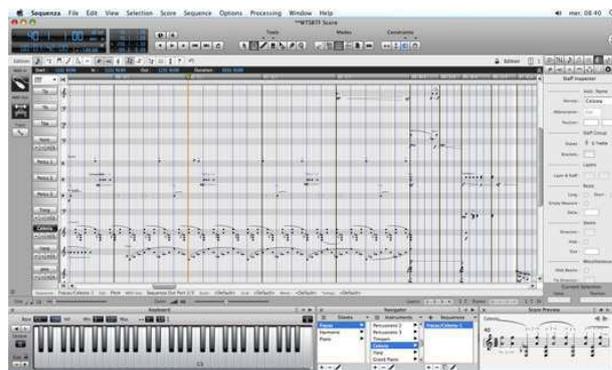


Figure 4. Mode de représentation proportionnel

MIDI » dans l'application), le temps est représenté de manière linéaire, un certain nombre de pixels équivalant à une certaine durée musicale. Deux rondes sont donc quatre fois plus espacées l'une de l'autre que deux noires. Dans ce mode, l'iconographie classique intervenant dans la notation du rythme n'est pas utilisée : toutes les notes sont noires, elles n'ont ni hampe ni ligature. Leur longueur est représentée par un trait de couleur dans une courbe dédiée. Lors du playback, le curseur de la tête de lecture se déplace donc à vitesse constante, au moins à tempo égal. Des discontinuités peuvent toutefois être introduites dans cet espace proportionnel pour ménager un espace graphique à certains signes qui en ont besoin : tout particulièrement les clés et les armures qui interviennent en cours de séquence. Ces espaces-là n'ont aucune durée temporelle : quand le curseur arrive à un endroit de ce genre, il le « saute » purement et simplement. Ce mode offre comme avantages, la simplicité, la vitesse d'exécution et la possibilité de représenter des longueurs temporelles importantes en une seule vue, permettant ainsi une appréhension visuelle des éléments formels.

3.2. Grilles

Pour découper le temps et aider à l'introduction de données rythmiques dans la séquence, Sequenza utilise le principe de la grille magnétique. C'est un principe bien connu, mais qui est ici poussé à l'extrême. Il existe dans Sequenza trois types de grilles, chacune étant entièrement configurable par l'utilisateur. On peut librement afficher ou masquer ces grilles, dans la règle ou dans la partition.

3.2.1. Mètre

La grille métrique exprime les signatures temporelles. Cette grille peut être remise en forme de diverses manières : édition interactive, menu édition (copier coller, etc.), éditeur et inspecteur spécialisés. Cette grille n'a aucune incidence sur la manière dont le matériel musical est organisé et stocké, on peut changer le mètre sans aucune incidence pour la séquence : à ce stade, le mètre n'est qu'un artifice graphique nous permettant de diviser le temps.

3.2.2. Grille rythmique

Quoique toutes les grilles soient potentiellement magnétiques, la grille rythmique est certainement la plus importante en termes de notation du rythme. En théorie, elle est constituée de valeurs plus petites que celle de la grille métrique (mais il n'y a là aucune obligation) que l'on dispose afin de permettre l'introduction de valeurs rythmiques rationnelles. La grille est d'abord constituée d'un simple pas que l'on peut librement définir (1/16 par défaut). On peut ensuite choisir une autre valeur (n'importe quelle valeur rationnelle) pour un segment donné de la ligne de temps et personnaliser ainsi la grille peu à peu. On peut aussi diviser un segment : par exemple, sélectionner une région de la longueur d'une noire et la diviser en trois – en pressant 3 – pour écrire ensuite un triolet.

Les valeurs de la grille sont utiles à production ou à la transformation de données, mais une fois les notes positionnées, on peut modifier la grille sans altérer la notation rythmique de la séquence. Tout comme la grille métrique, la grille rythmique dispose de nombreux outils d'édition.

3.2.3. Grille polyrythmique

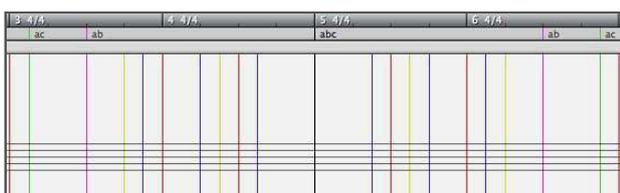


Figure 5. Un segment de grille polyrythmique (la mise en phase est au centre)

La grille polyrythmique offre la possibilité de déployer dans le temps un maximum de trois valeurs rythmiques indépendantes des deux autres grilles et, singulièrement, de la grille métrique. Ceci permet d'écrire des éléments temporels périodiques sortant du cadre de la mesure comme c'est souvent le cas dans l'écriture polyrythmique. Les trois valeurs sont librement définies, on peut aussi n'en utiliser qu'une ou deux. Il est également possible de déterminer où se situe leur première « mise en phase » (par défaut, à l'onset 0).

Graphiquement, chaque valeur est symbolisée par une couleur primaire, les points de rencontre entre deux valeurs sont symbolisés par la couleur complémentaire ré-

sultante, les points de rencontre entre les trois valeurs sont symbolisés par la couleur noire (voir figure 5).

3.2.4. Marqueurs temporels

À ces 3 types de grille s'ajoutent encore les marqueurs temporels qui permettent de marquer des points temporels non périodiques.

3.3. Tempo

Dans Sequenza, le tempo couvre une plage de 8 bits déployée entre 16 et 271. Afin de permettre la gestion de tempos plus rapides, un facteur multiplicateur est utilisé qui permet de multiplier le tempo de base par 2, 4 ou 8 permettant de fait l'utilisation d'un tempo montant jusqu'à 2168... Ceci nous mène d'ailleurs aux portes de la musique électronique, mais c'est un point que je ne développerai pas ici.

Les variations de tempo sont traitées dans des courbes qui sont représentées de la même manière que les courbes MIDI (volume, panoramique, etc.). Ces courbes permettent l'expression, au choix, d'une série de valeurs discrètes (comme dans le langage MIDI) ou d'une série de points reliés par des diagonales dont les valeurs discrètes sont calculées à la volée.

3.3.1. Processeurs liés au tempo

Le traitement de musique à tempos multiples superposés est une demande qui pose un véritable problème de cohérence quand on souhaite maintenir un lien avec une partition générale dont le principe est celui d'un tempo unique, possiblement battu par un chef. Sequenza propose plusieurs outils permettant de traiter le tempo indépendamment d'une piste à une autre sans pour autant briser l'unité temporelle nécessaire à la partition et au moteur temps réel.

Le processeur de « compression-extension temporelle » est un principe bien connu qui permet ici de traiter la plupart des objets de la partition, y compris ceux qui sont purement graphiques (comme les expressions). La façon dont le facteur de compression est contrôlé correspond particulièrement à la nécessité d'une musique polyrythmique à tempo multiple.

La « consolidation de tempo » permet d'appliquer une suite de changements de tempo dessinée dans une courbe aux valeurs rythmiques des accords de sorte que le rendu sonore reste identique une fois la courbe de tempo effacée (ceci induit une réorganisation des accords dans le temps). Il est ainsi possible de superposer plusieurs fluctuations de tempo complètement différentes dans des pistes distinctes (par exemple un *accelerando* et un *rallentendo* superposés et synchronisés).

3.4. Mode de notation classique

Comme présenté à la figure 6, le second mode de représentation de Sequenza est un mode de notation classique

présenté au format page.

Une des originalités de Sequenza est sa manière d'utiliser le système de stockage du temps musical pour générer une partition classique automatiquement. Il s'agit certainement du point sur lequel le travail de recherche et développement a été le plus important.



Figure 6. Mode de représentation classique

3.4.1. Codage de la position temporelle

Les positions temporelles des accords et des grilles de Sequenza sont stockées sous forme de fractions alors que, dans le langage MIDI, les valeurs temporelles sont stockées sous forme de « ticks » (The MIDI File Format, PPQN Clock) [12].

Certains logiciels codent leurs fichiers MIDI avec un nombre de ticks de 1024 par noire (2^{10}), ce qui est un choix cohérent d'un point de vue informatique, mais assez peu d'un point de vue musical car ce chiffre n'a que 2 comme facteur : un simple triolet ne peut être codé sans approximation. Historiquement, les constructeurs ont donc généralement privilégié des nombres qui ont au moins deux facteurs : 2 et 3. Ainsi 96, 192, 384 ou 768 ont été des standards selon les époques. « Protocols » utilise 960, un nombre qui présente l'avantage d'être également divisible par 5.

Quoi qu'il en soit, un programme qui doit pouvoir déduire une notation rythmique musicale incluant des groupes complexes sur base d'onsets temporels sans erreur d'approximation ne peut pas se baser sur un système de codage en ticks. Seul le codage d'un numérateur et d'un dénominateur comme deux nombres entiers permet de représenter des valeurs rythmiques sans aucune erreur d'approximation. C'est pourquoi Sequenza a fait ce choix.

3.4.2. Conversion en notation classique

Il n'est pas possible de donner ici une explication détaillée de l'algorithme qui permet à Sequenza de transcrire automatiquement une notation simplement proportionnelle en notation métrique. La complexité de cette procédure tient notamment à la multiplicité des interprétations possibles souvent dépendantes de la culture et de la tradition. Voici toutefois un exemple simple :

Imaginons une valeur temporelle rationnelle placée quelque part dans la ligne de temps. Il s'agira vraisemblablement d'un nombre difficilement lisible, mais on peut très facilement le simplifier en le rendant relatif au début de la mesure. Imaginons que ce nombre devienne alors $9/28$.

Le dénominateur 28 étant égal à $2^2 \cdot 7$, il en ressort que $1/28$ est la valeur de base d'un septolet de doubles-croches.

Le quotient du numérateur $9 : 7 = 1$ indique que ce septolet commence sur la seconde noire de la mesure et le reste ($9 \bmod 7 = 2$) indique que la valeur se situe sur le troisième pas du septolet.

L'analyse du numérateur nous indique toutefois que $9/28$ pourrait aussi être le quatrième pas d'un groupe de 7 croches divisant une blanche pointée commençant au début de la mesure.

Il faudra donc vérifier ces extrapolations à la lumière du contexte, des autres membres du groupe, et des marqueurs d'unités de la signature temporelle. Ceci se fera dans le flux de l'analyse séquentielle des valeurs rythmiques de la mesure (couche polyphonique par couche polyphonique), en réévaluant les extrapolations faites au fur et à mesure que l'analyse affine, confirme ou infirme les précédentes hypothèses.

Si l'analyse débouche sur une impossibilité, due à une trop grande complexité ou à une valeur irrationnelle, le groupe en cours est simplement aligné graphiquement sur les valeurs binaires les plus proches.

Le bon ajustement de cet algorithme nécessite la recherche d'une limite raisonnable à la complexité rythmique tolérée. En effet, l'ouverture à un niveau de complexité trop important peut fragiliser l'efficacité de l'interprétation de données simples ce qui tendrait à rendre l'application obscure dans ses résultats.

Sequenza limite donc ses recherches à tous les facteurs premiers inférieurs à 32.

4. RENDU GRAPHIQUE ET RENDU SONORE

Toute la problématique de la relation entre rendus graphique et sonore commence quand on rencontre des « gestes » musicaux qui ne s'expriment pas de la même manière dans les deux registres. Un trille qui s'écrit comme une simple valeur longue surmontée d'un signe dans la partition et qui se joue comme une série de valeurs brèves alternées. Un accord brisé qui s'écrit aligné verticalement au côté d'un signe graphique signalant que l'accord doit être « arpégé » alors que le rendu sonore nécessite l'écriture d'une série de notes non alignées. On pourrait multiplier les exemples, ils sont nombreux.

Face à cette problématique, Sequenza offre une gamme variée d'outils qui mettent en œuvre différentes stratégies. Leur combinaison permet de trouver une réponse adéquate à la plupart des problèmes posés. La complexité induite par ces artifices peut être compensée par une automatisa-

tion partielle des opérations.

4.1. Durée et durée graphique

Au sens du langage MIDI, la durée est simplement la distance en ticks qui sépare le message « Note On » (enfonce-ment de la touche) du message « Note Off » (relâche-ment de la touche). Chaque note possède donc sa propre durée.

D'une autre côté, au sens de la représentation dans la partition, chaque accord possède une durée qui est exprimée sous forme d'une valeur rationnelle binaire, possiblement contractée par le fait qu'elle se trouve dans un groupe irrégulier, possiblement allongée par le fait qu'elle soit pointée, et possiblement brisée en plusieurs valeurs rythmiques liées.

Ceci montre à quel point ces deux notions sont distinctes et à quel point le passage de l'une à l'autre demande un travail d'interprétation important.

Sequenza propose deux valeurs distinctes pour la durée. La première est l'équivalent de la durée MIDI, exprimée en ticks et stockée note à note : c'est la durée audible commune à tout séquenceur MIDI. La seconde, moins courante, est appelée « durée graphique » et est stockée dans l'accord. Cette valeur sert essentiellement à préciser la durée que l'on souhaite donner à un accord quand il est converti en notation classique.

4.2. Positionnement uniquement audible

Comme dans *Open Music*, chaque note possède une position temporelle purement sonore appelée « offset » qui est relative à son accord parent. Cette valeur peut changer le positionnement audible de la note de maximum une noire dans les deux directions du temps. Ceci permet, par exemple, d'écrire un accord qui est brisé du point de vue audible, mais qui reste aligné graphiquement.

L'offset est aussi utilisée pour offrir la possibilité d'aligner un phrasé rythmique (par exemple enregistré en temps réel) sur la grille de manière seulement graphique : cette valeur garde alors mémoire des deltas entre la « performance » et les valeurs alignées.

4.3. Accords invisibles et accords inaudibles

Les accords peuvent être déclarés comme étant inaudibles ou invisibles. Les accords invisibles restent visibles (quoique semi transparents, voir figure 7) en mode proportionnel, mais ne le sont plus du tout en mode de notation classique. De ce fait, ils sont exclus du calcul d'espace-ment et de mise en page. Ils restent cependant toujours audibles. Les accords inaudibles se transforment quant à eux en objets purement graphiques. Il est ainsi possible de dissocier à l'extrême ce qui est écrit de ce qui est joué.

La figure 7 donnent un exemple de l'utilisation des accords invisibles et inaudibles pour construire une geste musical dont les représentations sonore et graphique dif-fèrent largement. Les accords en vert pâle sont invisibles

en notation classique. Au contraire, les accords visibles en notation classique sont inaudibles.

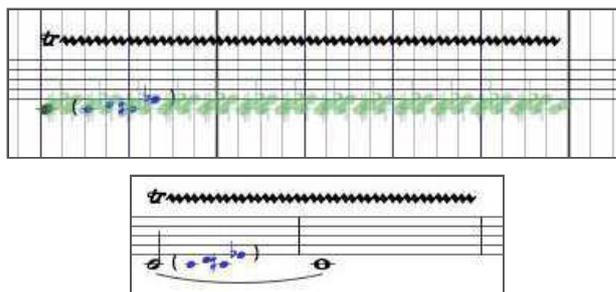


Figure 7. Un trille complexe en mode proportionnel, puis le résultat graphique en notation classique

4.4. Couches polyphoniques

Ce concept - qui s'apparente à celui de *calque* dans un éditeur graphique - est utilisé de par la plupart des logiciels de la famille des éditeurs de partitions. Dans Sequenza, les couches polyphoniques permettent d'écrire dans quatre couches indépendantes par portée, et il est possible de les régler pour qu'elles soient invisibles ou inaudibles. Il y a donc là une solution plus générale à la problématique abordée au point précédent.

5. ARCHITECTURE

Comme utilisateur d'informatique musicale, j'ai toujours trouvé frustrante l'organisation de la grande majorité des applications autour d'une ligne de temps unique ne comprenant de surcroît qu'un jeu de méta-événements (mètre et tempo)³. Cette conception ne me semble pas en phase avec le travail du compositeur qui est souvent constitué de brouillons, de recherches et d'essais multiples, de versions diverses, le tout s'ajoutant naturellement à la partition principale et constituant avec elle un ensemble intellectuel homogène. C'est pourquoi Sequenza est organisé très différemment.

5.1. Instrument et feuille

Au traditionnel empilement d'instruments, Sequenza ajoute une dimension : la « Feuille » (Sheet). Ce terme fait référence au travail sur papier dans lequel le compositeur travaille sur une partition générale, mais peut aussi avoir d'autres matériaux complémentaires sur des feuilles distinctes.

Autant l'instrument est un objet « horizontal » destiné à être superposé à d'autres dans la partition, autant la feuille peut être vue comme un objet « vertical » (parce qu'il « traverse » tous les instruments) destiné à en côtoyer

3 . Le terme de « méta-événement » fait ici référence à la norme MIDI et à son format de fichier. Tout événement qui n'est pas spécifiquement lié à une piste, mais concerne l'ensemble du morceau, est nommé « meta event » et doit être placé dans la piste zéro. On y trouve tout particulièrement les changements de tempo et les signatures temporelles.

d'autres dans une relation beaucoup moins déterminée. Les feuilles ne sont pas des « parties » car cela supposerait que ces objets soient destinés à s'enchaîner dans le temps, ce qui n'est pas le cas. Les feuilles sont plutôt le lieu où faire des essais divers, où décliner des versions, où construire une série de mouvements faisant partie d'un même ensemble, mais sans que cet ensemble ait nécessairement une organisation temporelle déterminée.

5.2. Cellule, séquences et héritage

Pour aider à la compréhension de ce qu'est une cellule, on peut prendre l'exemple d'un tableau dont les instruments seraient les rangées et les feuilles les colonnes⁴.

De la jonction entre une feuille et un instrument naît une cellule dont la seule fonction est d'être un réservoir de séquences (au minimum une). Chaque cellule, et chaque séquence-enfant, a donc deux parents, un instrument et une feuille.

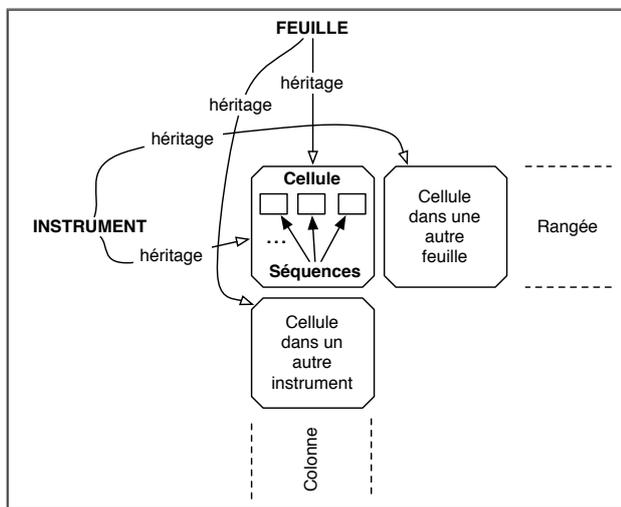


Figure 8. Feuille, Instrument, Cellules et Séquences

Autant l'instrument est le conteneur naturel des objets liés à la piste (système de portées, destination MIDI, mode microtonal, échelle, etc.), autant la feuille est le conteneur naturel des méta-événements³ (grilles, mètre, tempo, marqueurs temporels, etc.) qui affectent toutes les pistes.

Par défaut, chaque séquence hérite des propriétés de ses deux parents, mais peut ensuite « autonomiser » certaines d'entre elles en se les attachant directement. Ce système d'héritage donne à Sequenza une grande souplesse de fonctionnement, partant d'une situation simple où les séquences partagent toutes les propriétés de leurs parents et évoluant, quand nécessaire, vers une situation plus sophistiquée où chaque séquence possède ses propres propriétés.

4. Si, dans ce tableau, l'axe des y représente la superposition des instrument dans la partition, l'axe des x ne représente pas le temps : c'est plutôt une abstraction liée à l'organisation des données (comme, par exemple, un onglet dans un navigateur).

5.3. Navigation



Figure 9. Navigation dans la structure d'une session

Le navigateur de Sequenza, présenté à la figure 9 permet la navigation dans la structure de la session. Il adopte une présentation proche de celle d'une arborescence de fichier. La cellule courante est représentée par le réservoir de séquences qui se trouve à la droite de la vue.

5.4. Feuille et multiséquence

Chaque feuille contient une ou plusieurs partitions poly-instrumentales appelée multiséquence. Comme la séquence simple, la multiséquence hérite ses méta-événements de la feuille parente et peut aussi s'attacher en propre un ou plusieurs types de méta-événements.

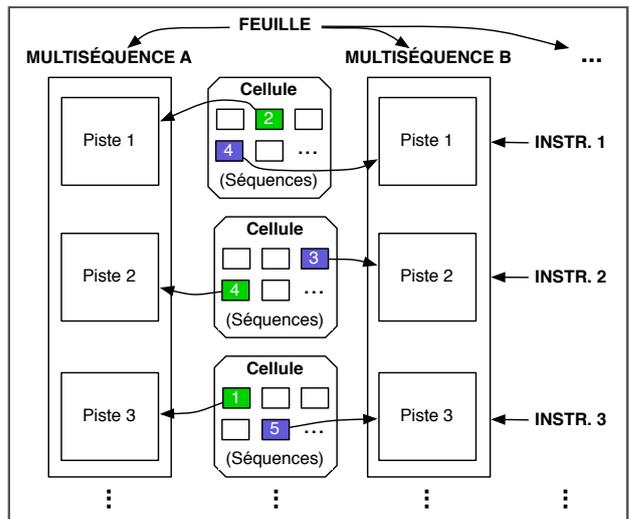


Figure 10. Organisation des multiséquences dans une feuille : De même que chaque cellule comporte un nombre illimité de séquences, chaque feuille comporte un nombre illimité de multiséquences (par défaut une seule)

Une multiséquence contient N séquences, une pour chaque piste. Comme on le voit à la figure 10, dans chaque multiséquence la piste peut utiliser l'une des séquences disponibles dans le réservoir de séquences que constitue la cellule.

5.5. Feuille et multi-instrumentation

Afin de ne pas être limité par le caractère « rectangulaire » du tableau évoqué au point 5.2, chaque

feuille peut ne pas utiliser tous les instruments disponibles dans la session. Il est ainsi possible - quoique la session ne comporte qu'un seul jeu d'instruments - d'avoir des effectifs instrumentaux partiellement ou même totalement différents dans les différentes feuilles.

Par exemple, pour réaliser une pièce pour orchestre dans une feuille et sa réduction pour piano dans une autre, il suffit que la session comporte un effectif instrumental constitué d'un orchestre plus un piano. Le piano sera désactivé dans la première feuille, et l'orchestre sera désactivé dans la seconde feuille.

5.6. Parties individuelles

Enfin, pour travailler sur une partie individuelle (par exemple pour la production d'un matériel orchestral), il suffit de passer du mode de représentation poly-instrumental (où la multiséquence est l'objet représenté dans la partition) au mode mono-instrumental (où la séquence est l'objet représenté dans la partition) ⁵.

6. PARTAGE

Sequenza se veut le plus ouvert possible au partage de ses données afin de pouvoir s'intégrer harmonieusement à un ensemble logiciel. Voici un rapide tour d'horizon des moyens par lesquels Sequenza communique avec le monde extérieur :

MIDI File (import et export des données MIDI)

xmlMusic (export graphique vers un éditeur de partition)

PDF (export graphique)

Wave et Aiff (export audio)

Mov, Mp4, Avi, M4v, Dv (import multimédia)

SDIF (import d'analyses sonores spectrographiques)

SF2 (import de bibliothèques sonores)



Figure 11. Travail avec fichier vidéo au format QuickTime

5. Le concept de mode mono-instrumental ou poly-instrumental est totalement indépendant du concept de feuille : On peut passer d'un mode à l'autre indépendamment du fait que l'on soit dans l'une ou l'autre feuille. Le mode poly-instrumental utilise toute la « colonne » tandis que le mode mono-instrumental n'utilise que l'une de ses cellules.

6.1. Ports de communication temps réel

Sequenza communique en temps réel au format MIDI et peut créer un nombre illimité de ports de sortie virtuels. Les données entrantes sont aiguillées vers le port actif (MIDI bypass) où elles peuvent aussi être enregistrées. Actuellement, le nombre de types d'événements MIDI interprétés est toutefois limité.

Sequenza dispose également d'un clavier virtuel sophistiqué qui offre des fonctionnalités originales pour la commande des données MIDI via le clavier alphanumérique.

6.2. Plug-ins

Sequenza héberge et commande tout type d'instrument virtuel compatible avec Mac OS X. Les données des plug-ins (AUPreset) sont préservées dans la session elle-même.

6.3. Scriptage

Sequenza possède un système de scriptage qui permet à l'utilisateur de générer ou de transformer de l'information dans la partition. Le langage de script utilisé est orienté vers la production de séries de valeurs qui représentent des hauteurs, des rythmes, des durées ou des dynamiques. L'organisation des scripts en modules et chaînes de modules ressemble assez à une forme de synthèse sonore dont le résultat serait, non pas un signal audio, mais de la musique écrite.

Les scripts sont organisés en « Patterns » qui peuvent être sauvegardés dans une banque et exportés dans la partition en cliquant simplement sur un bouton, ou en utilisant un mode spécifique de l'outil crayon.

Il est aussi possible d'importer de la musique dans l'éditeur, c'est-à-dire de convertir automatiquement un segment de partition en script, afin de pouvoir la manipuler.

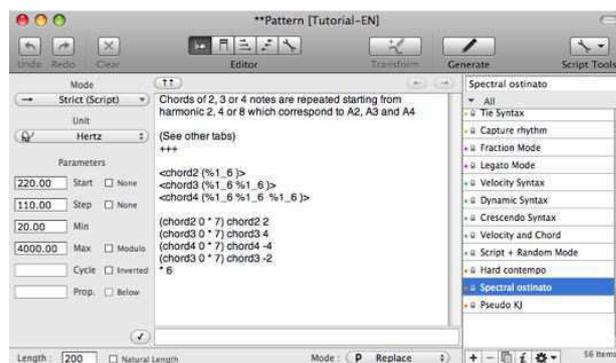


Figure 12. La fenêtre de scriptage

À moyen terme Sequenza devrait aussi pouvoir être contrôlé via *AppleScript*.

7. CONCLUSION

Sequenza tente d'apporter une contribution à tous ceux qui travaillent aujourd'hui à la reconstruction pour notre

musique, sinon d'un langage commun, au moins d'un espace conceptuel partagé, tout particulièrement en s'appuyant sur les techniques, si riches et si nombreuses, que nous a légué le XXe siècle. Il m'a toujours semblé que l'une des responsabilités de notre génération était de consolider les nombreux concepts élaborés durant ce siècle en les intégrant pleinement à une théorie musicale modernisée et en leur donnant toutes sortes de prolongations constructives.

C'est aussi à cette lumière qu'il faut lire le caractère parfois « vulgarisateur » du projet Sequenza. Tout en maniant des concepts parfois ardu de la théorie musicale classique, Sequenza se veut une porte ouverte vers ces savoirs pour ceux, nombreux, qui venant d'autres espaces culturels en sont trop souvent intimidés.

La plupart des fonctionnalités examinées ici illustrent cette volonté de débarrasser l'utilisateur d'opérations souvent trop fastidieuses - voire décourageantes - dès que l'on touche à ces questions que l'industrie considère sans doute comme marginales : groupes complexes, microtonalité, modes non-octavians, polyrythmes, etc.

Pour le compositeur débutant, le fait que ces manipulations soient plus simples est une opportunité de s'ouvrir à ces concepts, d'en comprendre l'intérêt créatif, et de les intégrer plus aisément à son vocabulaire.

Pour le compositeur aguerri, cette mise sur un pied d'égalité des différentes techniques d'écriture, récentes ou plus anciennes, permet une clarification du langage et un gain de temps dans l'exécution des tâches. Ainsi, l'attention peut être plus largement orientée vers la dimension créative du travail de composition.

8. REMERCIEMENTS

Je souhaite remercier ici tous ceux qui m'ont aidé à porter le projet Sequenza :

L'équipe de développement d'*Open Music* dont travail m'a inspiré et aidé, et tout particulièrement Gérard Assayag, qui m'a donné de nombreux conseils au cours de la première phase de développement.

L'équipe d'ingénieurs de Lispworks ltd, et tout particulièrement Martin Simmons, et la communauté d'utilisateurs de *LispWorks*, qui m'ont apporté un soutien précieux au fil de ces années.

L'équipe d'ingénieurs du *CETIC*, et tout particulièrement Christophe Ponsard, qui m'a véritablement ouvert l'esprit en matière d'architecture logicielle au cours de la dernière phase de développement. Le *CETIC* héberge aujourd'hui gracieusement les codes de Sequenza dans sa forge.

Tous les bêta-testeurs, compositeurs et amis qui m'ont aidé à réévaluer, à améliorer et à corriger Sequenza, lors de ses phases alpha et bêta, notamment Stéphane Roumieux, Maya Tell-Nohet, ainsi que les étudiants de ma classe de *Composition en Musiques Appliquées et Interactives* du Conservatoire royal de Mons qui ont bien voulu s'impliquer dans cette phase de mise au point.

Je remercie enfin Irène Deliège et Jean-Luc Fafchamps pour leur relecture amicale et leurs conseils avisés.

9. REFERENCES

- [1] LispWorks Ltd. *Objective-C and Cocoa User Guide and Reference Manual*, 2008.
- [2] MIDI Manufacturers Association *MIDI 1.0 Specification*, 1982.
- [3] Apple Inc. *Audio and MIDI on Mac OS X* (p36) 2001.
- [4] OM 6.5 *User Manual, section MIDI 3.e Microintervals*, IRCAM. <<http://support.ircam.fr/forum-ol-doc/om/om6-manual/co/Microintervals.html>>
- [5] MIDI Manufacturers Association *MIDI Manufacturers Investigate HD Protocol*, 2008-2012.
- [6] MIDI Manufacturers Association *MIDI Tuning Messages*, 2008-2012, 1995-2012. <<http://www.midi.org/techspecs/midituning.php>>
- [7] Letz S., Fober D., Orlarey Y. *Modèles de structures tonales dans Elody*. GRAME / JIM, 1995.
- [8] Rasch R. *The Unification of Tonal System, or About The Circle of Fifths* (p 320 & 326), Journal of New Music Research, 2000.
- [9] Hakim N. et Dufourcet M-B. *Guide Pratique d'Analyse Musicale* (p 110). Éditions Combre, Paris, 1995.
- [10] Slonimsky N. *Thesaurus of Scales and Melodic Patterns* (p ii), Amsco Publication, New York, 1947.
- [11] OM 6.5 *User Manual, section Score Objects 1.2 Time Representation*, IRCAM. <<http://support.ircam.fr/forum-ol-doc/om/om6-manual/co/Score-Objects-Intro.html#aeNe6>>
- [12] MIDI Manufacturers Association *MIDI File Specification - PPQN Clock*, 1988.