

ÉCOLE NATIONALE SUPÉRIEURE LOUIS LUMIÈRE
La Cité du Cinéma, 20 rue Ampère, BP 12, 93213 La Plaine Saint-Denis
Tél. : +33 (0) 1 84 67 00 01 — www.ens-louis-lumiere.fr

MÉMOIRE DE FIN D'ÉTUDES

2018

ÉTUDE DE L'INTERFACE ENTRE UNE STATION DE TRAVAIL AUDIONUMÉRIQUE ET LA PARTITION

**Optimisation de la chaîne de composition et de
production d'une musique de film**

**Partie pratique :
Développement d'un outil logiciel
de quantification adaptative**

Nelson SANTONI

Sous la direction de
Jean ROUCHOUSE, directeur interne
Sylvain MORIZET, directeur externe

Membres du jury
Gérard PELÉ, président du jury
Sylvain LAMBINET, rapporteur

Remerciements

Je voudrais tout d'abord remercier mes deux directeurs de recherche, Jean Rouchouse et Sylvain Morizet, qui m'ont accompagné avec patience et bienveillance tout au long de la rédaction de mon mémoire et du développement de ma partie pratique.

Je remercie également Nelson Malléus et Damien Salançon, qui au cours de mes stages puis des divers échanges que nous avons eus m'ont beaucoup appris sur les réalités du terrain.

Merci à tous mes camarades, aux côtés de qui ces trois années sont passées comme un éclair. J'espère qu'ils continueront à m'accompagner pour de longues années encore.

Merci à mes professeurs de l'ENS Louis-Lumière, grâce à qui j'appréhende avec sérénité mon entrée dans le monde professionnel.

Merci à Sylvain Lambinet, dont les conseils, l'expertise et l'affection ont largement contribué à mon épanouissement.

Merci à ma famille, et tout particulièrement à mes parents pour leur soutien inébranlable et la confiance qu'il m'ont toujours témoignée.

Merci à Viviane, qui n'a cessé de m'encourager avec tendresse et enthousiasme.

Enfin, mes pensées vont à Jean Santoni, mon « troisième grand-père », avec qui j'aurais dû partager ces derniers instants étudiants et la perspective d'un avenir enthousiaste, mais qui depuis le mois d'avril repose à Sète, au bord de la mer Méditerranée.

Résumé

À l'instar des progrès techniques qui ont jalonné l'industrie cinématographique, l'informatique musicale a révolutionné les méthodes de composition et de production de la musique de film. Aujourd'hui, l'environnement et le flux de travail du compositeur moderne s'articulent autour d'une STAN (Station de Travail Audio-Numérique), grâce à laquelle il compose, enregistre, produit, et surtout maquette. En effet, depuis quelques années, les instruments virtuels offrent la possibilité de produire des maquettes vraisemblables, qui ressemblent à s'y méprendre à la musique définitive qui sera enregistrée par de véritables musiciens. Fort heureusement, il n'est pas encore d'actualité de négliger et d'abandonner les bénéfices d'une séance d'enregistrement, et de ce fait, il est nécessaire d'orchestrer et d'éditer des partitions d'après la maquette produite par le compositeur.

Ainsi, une fois que la maquette programmée par le compositeur est prête, elle est exportée dans un fichier MIDI standard, qui est à son tour importé dans un logiciel de notation musicale. C'est dans ce logiciel que la maquette est orchestrée et que les partitions sont éditées à l'attention des musiciens.

Compte-tenu de l'incessante réduction des temps de production et des budgets alloués par l'industrie cinématographique, les développeurs et les utilisateurs de leurs outils sont dans une recherche constante et fiévreuse de l'optimisation et de la fluidité de l'ensemble du processus. Si les rouages du maquettage, de l'orchestration et de la copie sont bien huilés, ce n'est pas le cas du passage de la STAN au logiciel de notation musicale. Cette interface entre le monde de la STAN et celui de la partition ne fonctionne pas à merveille, et mérite que nous nous y attardions ; il ne suffit que d'une goutte d'huile pour que tous les maillons de la chaîne fonctionnent de concert.

Mots-clés : musique de film, MIDI, STAN, notation musicale, partition, quantification, maquette, instrument virtuel.

Abstract

Just as the technical improvements have punctuated the film production industry, computer music has revolutionized the way film music is composed and produced. Nowadays, the modern scoring film composer's workflow is structured around a DAW (Digital Audio Workstation), which allows him to compose, record, produce, but most of all to mock his music up. Indeed, over the last years, virtual instruments have offered the possibility to produce credible mock-ups, which sound like the final music recorded by real musicians. Luckily, it is not topical to do without the benefits of a recording session, that is why it is necessary to orchestrate and edit music sheets according to the composer's mock-up.

Thereby, once the music mocked up by the composer is ready, it is exported to a Standard MIDI File (SMF), which is then imported into a music scoring and notation software. It will then be orchestrated, and music sheets will be edited for the attention of musicians.

Given the fact that production schedules and budgets granted by the film producers are decreasing, the developers and the users of the tools are constantly looking for ways to optimize and make the entire scoring process more fluid. If the machinery of mock-upping, orchestrating and music copying is well-oiled, switching from the DAW to the music scoring and notation software does not work like a charm, and it deserves our attention; it may take a single drop of oil in order to make all the scoring chain act in concert.

Keywords : film music, MIDI, DAW, music notation, scoring, quantization, mock-up, virtual instrument.

Table des matières

Remerciements.....	i
Résumé.....	ii
Abstract.....	iii
Table des matières.....	iv
Introduction.....	1
I. La musique au cinéma : histoire et pratiques.....	3
A. Histoire brève de la musique de film.....	3
1. Les films muets.....	5
2. Les premiers films parlants : les débuts de la musique synchronisée.....	8
3. Le système des « studios » hollywoodiens.....	9
4. L'arrivée de la télévision et la fin du système des studios.....	12
5. De la fin des studios à aujourd'hui : la révolution numérique.....	14
B. Traditions et évolutions des pratiques de composition de musique à l'image.....	17
1. Les premières méthodes de travail.....	17
2. L'arrivée des synthétiseurs et de l'informatique.....	18
3. Les outils de travail actuels.....	20
C. La fabrication d'une musique de film.....	23
1. Rappels : la chaîne de fabrication d'un film.....	23
1.1. La pré-production.....	24
1.2. La production.....	25
1.3. La post-production.....	25
2. Composition de la musique d'un film : terminologie et travaux préliminaires.....	28
3. La composition, l'orchestration, et la préparation de la session d'enregistrement de la bande originale.....	33
4. La séance d'enregistrement et le mixage de la musique.....	38

II. Le studio de composition : le MIDI et les instruments

virtuels.....41

A. Le MIDI : rappels et pré-requis.....	41
1. Petite histoire du MIDI.....	41
2. La structure d'un fichier MIDI (SMF).....	42
2.1. Le « <i>Header Chunk</i> » ou le bloc d'en-tête.....	45
2.1.1. Les formats d'un fichier MIDI standard.....	46
a. Le Format 0.....	46
b. Le Format 1.....	46
c. Le Format 2.....	47
2.1.2. Le nombre de pistes <ntrks> d'un fichier MIDI.....	47
2.1.3. La <division> temporelle d'un fichier MIDI.....	47
2.2. Le(s) « <i>Track Chunk(s)</i> » ou le(s) bloc(s) de piste.....	50
2.2.1. Le <delta time> ou Δt	51
2.2.2. La quantité de longueur variable (VLQ).....	51
2.2.3. Les trois types d'évènements d'un bloc de piste.....	53
a. Les <i>MIDI Channel Events</i> (0x8n - 0xEh).....	53
b. Les évènements SysEx (0xF0 et 0xF7).....	54
c. Les évènements <i>méta</i> (0xFF).....	55
3. Utilisation du MIDI dans une STAN.....	58
3.1. Définitions et fonctionnalités.....	59
3.2. Les différences entre un flux de données et un fichier MIDI.....	61
3.3. Les pistes « MIDI » et les pistes « Instrument ».....	62
B. Les instruments virtuels et leur utilisation.....	63
1. Les différentes familles d'instruments virtuels.....	63
2. La gestion et l'utilisation d'un studio hybride.....	65
2.1. La cohabitation entre le physique et le virtuel.....	66
2.2. La construction de <i>template(s)</i>	68
2.3. <i>Vienna Ensemble PRO (VSL)</i>	70
3. Enregistrement et production d'une maquette MIDI.....	73
3.1. Gestion des articulations et des modes de jeu.....	74
3.2. Gestion des nuances et de la dynamique.....	80
3.3. Pré-mixage et finalisation.....	82

C. Avantages et inconvénients des outils modernes.....	86
1. Similarités et spécificités de l'orchestration MIDI.....	86
2. L'humanisation, ou la recherche de la vraisemblance musicale.....	89
3. La quantification.....	93

III. Développement d'un outil logiciel pour optimiser le passage de la STAN à la partition.....96

A. Transfert de données d'une STAN vers <i>Sibelius</i> : les bases d'une interface plus adaptée.....	96
1. La <i>liste d'évènements</i> ou « <i>Event List</i> ».....	96
2. Import d'un SMF dans <i>Sibelius</i>	98
3. Conservation d'un socle commun de développement.....	101
B. Proposition d'un cahier des charges fonctionnel.....	102
1. Informations connues, données, ou déduites.....	103
1.1. Informations globales issues du SMF.....	103
1.2. Informations données par l'utilisateur.....	104
1.3. Informations déduites.....	105
a. Détermination du système de numération rythmique.....	105
b. Création d'une liste d'évènements.....	107
c. Fabrication de tables de référence à l'échelle de la mesure et du temps musical	109
d. Détermination et restriction des valeurs rythmiques possibles.....	112
2. Création de sous-listes d'évènements.....	116
2.1. Répartition des évènements par mesure.....	117
2.2. Répartition des évènements par temps.....	117
2.3. Calcul du nombre d'évènements par mesure et par temps.....	118
3. Quantification adaptative des données.....	120
3.1. Quantification de la position de début de chaque note.....	120
a. Réduction de la fenêtre d'analyse.....	121
b. Choix d'un pas de quantification.....	122
c. Analyse et modification des données brutes.....	123
3.2. Quantification de la longueur de chaque note.....	125
3.3. Vérification et correction éventuelle des données.....	127
4. Reconstitution d'un SMF quantifié.....	128

C. Vers une interface transparente : quelques perspectives de développement et d'améliorations.....	131
1. Gestion des instruments polyphoniques et des articulations.....	131
2. Préparation des données pour les importer dans une <i>template</i>	132
3. Analyses croisées et corrélations musicales.....	133
Conclusion.....	135
Bibliographie.....	138
Sites web.....	139
Annexe 1 : Lexique et définition des termes musicaux les plus utiles à la compréhension de notre mémoire.....	140
Annexe 2 : Quelques exemples de nombres entiers codés en VLQ donnés par la spécification du SMF.....	143
Annexe 3 : Liste des contrôles continus (CC) les plus utilisés dans le cadre de la composition de musique à l'image.....	144
Annexe 4 : Algorithme de quantification du Δt_{abs} d'une <i>Note(x)</i> donnée.....	145
Annexe 5 : Algorithme de quantification de la longueur d'une <i>Note(x)</i> donnée.....	146

Introduction

Tout comme le cinéma et son industrie, la musique de film a connu de nombreuses évolutions au cours de son existence, aussi bien sur le plan de sa fabrication que de son esthétique. Nous nous proposons d'étudier ici la chaîne de production d'une musique de film, afin d'entrevoir et de comprendre les origines des méthodes et des outils des compositeurs modernes. En effet, la composition de musique à l'image est étroitement liée aux outils et aux technologies qu'elle exploite ; elle entretient même à son endroit une forme de dépendance.

Aujourd'hui, l'ensemble de l'industrie cinématographique est en passe d'achever définitivement son basculement vers le tout-numérique : ainsi, l'environnement de travail du compositeur moderne est globalement informatique. De plus, si le matériel physique et logiciel a énormément évolué depuis les balbutiements de l'informatique musicale dans les années 1970, l'une de ses fondations, elle, demeure toujours d'actualité : le *Musical Instrument Digital Interface* (MIDI)¹. Développé au début des années 1980, cette norme s'est rapidement imposée comme la norme de communication entre diverses « machines », comme des synthétiseurs, des contrôleurs ou des logiciels informatiques, mais aussi comme le standard de transfert de fichiers contenant des données musicales. En d'autres termes, la norme MIDI est aujourd'hui encore au cœur de l'environnement de travail du compositeur.

De ce fait, il cohabite avec des éléments beaucoup plus récents, et technologiquement plus avancés. Il n'en demeure pas moins que la normalisation du MIDI a dès le départ prévu une structure et une architecture tournées vers l'évolution et l'ouverture. Cela a permis au MIDI d'être mis à jour et de s'adapter aux évolutions technologiques successives. Cependant, dans certains cas, quelques-unes de ses fonctionnalités ont été détournées, ou utilisées à des fins qui n'étaient pas initialement prévues. Par conséquent, le

1 Nous pouvons traduire MIDI par « Interface Numérique des Instruments de Musique ».

transport de données MIDI entre plusieurs logiciels, par exemple entre une station de travail audionumérique (STAN) et un logiciel de notation musicale, peut présenter quelques difficultés. Des fonctions détournées dans le premier seront interprétées comme relevant de la fonction initiale dans le second.

Si le MIDI est au carrefour de l'échange de données musicales, il constitue aussi le socle de développement et d'exploitation des instruments virtuels, dont l'avènement et la prolifération sont sans aucun doute à l'origine de la dernière mutation significative du paysage de la musique de film en date. Il s'agit *grosso modo* d'instruments de musique logiciels qui produisent des sons électroniques ou cherchent à recréer le son d'instruments réels. Ils permettent dans les faits de créer des morceaux de musique sans avoir recours à de véritables musiciens. Tels qu'ils existent aujourd'hui, ces instruments virtuels sont « joués » et programmés grâce au protocole MIDI.

La méthode la plus répandue de composition de musique à l'image consiste donc aujourd'hui à produire des maquettes grâce à des instruments virtuels, puis à transférer cette maquette dans un logiciel de notation musicale, afin de l'orchestrer et d'éditer les partitions nécessaires à l'enregistrement de la version finale de la musique. Bien évidemment, les acteurs de l'industrie musicale cherchent — et ont toujours cherché — à optimiser cette chaîne de production, de la composition à la livraison de la musique finale. Cependant, un maillon semble encore gêner la fluidification totale du processus : le passage de la STAN au logiciel d'édition et de gravure de partitions.

Ainsi, une étude historique de la musique de film, au-travers du prisme de la méthodologie et des évolutions technologiques, puis la dissection de l'environnement de travail du compositeur moderne, nous permettront de comprendre pourquoi le transfert de données MIDI entre la STAN et la partition n'est pas fluide. Il sera alors possible de suggérer une solution pour pallier ce problème, sous la forme d'un outil logiciel.

I. La musique au cinéma : histoire et pratiques

A. Histoire brève de la musique de film

L'histoire de la musique de film est par essence conjointe à l'histoire du Cinéma. De ce fait, elle est plurielle, et localisée géographiquement. Dans le cadre de ce mémoire, c'est l'histoire technologique du cinéma qui doit nous guider dans l'appréhension du fonctionnement de la chaîne de production d'une musique de film. Nous ne prétendons donc pas proposer ici une histoire exhaustive de la musique de film : il s'agit plutôt de la parcourir au gré des évolutions techniques, technologiques, sociétales et économiques qui la jalonnent.

Si la France a connu, avec les frères Lumière, la naissance du cinéma, et qu'elle représentait, selon un rapport de l'INHESJ², 85 % des parts de marché dans le monde avant la Première Guerre mondiale, il n'en demeure pas moins que le cinéma hollywoodien s'est imposé, si bien qu'aujourd'hui « nous sommes incapables de concurrencer leur savoir-faire en la matière »³. Par ailleurs, comme l'écrit André Malraux dans son *Esquisse d'une psychologie du cinéma*, publiée en 1940, « le cinéma est une industrie », que nous pouvons peut-être davantage envisager comme un art industriel plutôt qu'une industrie culturelle. En d'autres termes, le cinéma est un art qui nécessite des outils de production industriels.

En 1895, donc, le « Cinématographe », concurrent direct du « Kinétoscope » d'Edison, est inventé par Auguste et Louis Lumière. Le 28 décembre de cette même année, les deux frères organisent à Paris la première projection cinématographique publique et payante : le cinéma « populaire de masse » est né. Assez rapidement, Georges Méliès (1861-1938), puis Charles

2 INHESJ : *Institut National des Hautes Études de la Sécurité et de la Justice*.

3 Groupe de veille et d'analyse « MALRAUX », *Influence culturelle d'Hollywood*, Paris, INHESJ, 2015, p. 40.

Pathé (1863-1957), ou encore Léon Gaumont (1864-1946), créent les premiers grands studios de cinéma français. Malheureusement, la Grande Guerre rompt cette prospérité artistique et économique, et la suprématie du cinéma français est peu à peu estompée par les États-Unis d'Amérique, qui prennent une place prépondérante dans l'industrie cinématographique, aussi bien en termes de production que de progrès techniques et technologiques.

En 1914, alors que le *Edison Trust* domine l'économie cinématographique de la côte Est des États-Unis, plusieurs compagnies décident de fuir ce monopole et s'installent sur la côte Ouest, à Hollywood. Quinze ans plus tard, 80 % des films américains sont produits sur la côte Est.

Le système hollywoodien est au cœur de notre étude, puisqu'il a toujours développé, et constitue encore aujourd'hui, une industrie qui réfléchit constamment à l'organisation et à l'optimisation de la chaîne de production d'un film. Le système des studios, puis l'organisation syndicale de l'industrie cinématographique hollywoodienne, ont rapidement induit un cloisonnement de la chaîne de production : cette hiérarchisation permet de cerner plus clairement les maillons qui la composent. Les métiers du cinéma à Hollywood sont cloisonnés et démultipliés : individuellement, ils correspondent presque à chaque tâche qui doit être accomplie. Comme le souligne le rapport de l'INHESJ, les États-Unis « ont été les premiers à standardiser la création cinématographique, et actuellement, le cinéma américain règne en maître sur l'ensemble de l'industrie cinématographique internationale »⁴. C'est pour toutes ces raisons que notre histoire brève de la musique de film sera centrée sur les États-Unis d'Amérique, et plus précisément sur Hollywood.

4 Groupe de veille et d'analyse « MALRAUX », *op. cit.*, p.19.

1. Les films muets

Il semble que les premiers films de l'Histoire aient été accompagnés de musique : en effet, le pianiste Émile Maraval apparaît dans le programme d'une projection organisée par les frères Lumière au mois de décembre 1895 à Paris, tandis que la première projection de Londres, qui a lieu le 20 février 1896 à l'Institut Polytechnique, est accompagnée par un harmonium ; les représentations londoniennes suivantes aux théâtres *Alhambra* et *Empire* sont même accompagnées par un orchestre⁵. Quoiqu'il en soit, ces projections rencontrent un vif succès auprès du public et, rapidement, diverses formations instrumentales accompagnent presque systématiquement, du moins dans la mesure du possible, toute projection cinématographique. La musique n'a cependant pas encore le rôle dramatique et narratif qu'elle acquiert au fil des années, et revêt alors seulement un rôle d'accompagnement et d'animation : elle n'est pas composée spécifiquement pour un film, et provient de toutes sortes de répertoires, de la chanson populaire aux « grands » compositeurs classiques, au sens large du terme.

En 1908, Camille Saint-Saëns (1835-1921) compose ce qui est aujourd'hui considéré comme la première musique « originale » écrite spécifiquement pour un film, *L'Assassinat du Duc de Guise*. Si la partition de Saint-Saëns plaît, le principe d'une partition spécifique à un film ne séduit pas la profession : le coût d'une telle production est bien trop élevé par rapport aux pratiques de l'époque puisque, outre la rémunération du compositeur, il est nécessaire d'engager un ensemble instrumental spécifique. Or, au début du XX^{ème} siècle, il y a autant d'ensembles instrumentaux qu'il y a de lieux de projection. Toutefois, un certain nombre d'acteurs de l'industrie cinématographique ont rapidement la conviction que s'il n'est pas [encore] viable économiquement de composer pour un film et un ensemble spécifiques, la musique qui accompagne les films peut être standardisée : c'est l'avènement des « recueils de thèmes », ou « *Music Fake Books* ».

5 James Wierzbicki, *Film Music: A History*, New York, Routledge, 2009, pp. 19-20.

Dans les années 1900, un certain nombre d'éditeurs commencent donc à proposer des recueils de thèmes classés par ambiance ou « humeur » musicales. Ces thèmes sont majoritairement issus du répertoire Classique et Romantique, ce qui ne les empêche pas de piocher aussi dans des morceaux plus populaires et plus récents. Chaque salle de projection qui en a les moyens emploie alors un directeur musical, qui après avoir déterminé l'ambiance musicale de chaque scène d'un film, sélectionne au sein de ces recueils les thèmes qu'il juge les plus pertinents, et les arrange pour l'ensemble instrumental à sa disposition. La plupart du temps, il compose de courtes transitions entre ces différents moments musicaux : nous pouvons concevoir aisément qu'un brusque changement d'armure, de *tempo*⁶, d'instrumentation, ou simplement d'humeur peut paraître étrange, voire hasardeux. Ces transitions fluidifient donc le passage d'une scène à l'autre.

Assez rapidement, Max Winkler propose un système plus avancé que celui des recueils de thèmes : il pense qu'en visionnant un film avant sa sortie, il peut proposer ce qu'il appelle des « *cue sheets*⁷ », c'est-à-dire une sélection de thèmes pour chaque scène, qu'il publie en même temps que la sortie du film pour que chaque salle de projection puisse l'accompagner de la même musique. Winkler ne propose par contre pas d'arrangement dans ses *cue sheets*, afin que chaque directeur musical puisse l'adapter à son ensemble instrumental. Quoiqu'il en soit, la méthode de Winkler permet d'aller plus loin dans la standardisation de la musique de film, puisque d'un lieu de projection à l'autre, la même musique est jouée pour accompagner un même film. À titre d'exemple, la *cue sheet* rédigée par Winkler durant la nuit où l'idée lui est venue est reproduite ci-après⁸.

6 L'Annexe 1 propose un lexique et quelques définitions sommaires des termes musicaux les plus utiles à la compréhension de ce mémoire.

7 Nous verrons tout au long de ce mémoire que les termes « *cue sheet* », que nous pouvons traduire par « feuille de montage », peuvent désigner divers documents, et donc avoir des significations différentes en fonction du contexte dans lequel ils sont employés. Ici, un *cue* désigne un extrait ou un fragment musical.

**Music Cue Sheet for
The Magic Valley
Selected and compiled by M. Winkler**

Cue

1. Opening — play *Minuet No. 2 in G* by Beethoven for ninety seconds until title on screen "Follow me dear."
2. Play — "Dramatic Andante" by Vely for two minutes and ten seconds. Note: play soft during scene where mother enters. Play Cue No. 2 until scene "hero leaving room".
3. Play — "Love Theme" by Lorenze for one minute and twenty seconds. Note: Play soft and slow during conversations until title on screen "There they go."
4. Play — "Stampede" by Simon for fifty-five seconds. Note: Play fast and decrease or increase speed of gallop in accordance with action on the screen."

Copyright © Carl Fischer

**Feuille d'extraits musicaux pour
La Vallée magique
Sélectionnés et compilés par M. Winkler**

Extraits

1. Ouverture — Jouer *Menuet No. 2 en sol majeur* de Beethoven pendant quatre-vingt-dix secondes jusqu'à l'apparition du titre « *Follow me dear* ».
2. Jouer — « Andante dramatique » de Vely pendant deux minutes et dix secondes. Note : Jouer doucement pendant la scène où la mère entre. Jouer le l'Extrait No. 2 jusqu'à la scène où « le héros quitte la pièce ».
3. Jouer — « Thème d'amour » de Lorenze pendant une minute et vingt secondes. Note : Jouer doucement durant les conversations jusqu'à l'apparition du titre « *There they go* ».
4. Jouer — « Cavalcade » de Simon pendant cinquante-cinq secondes. Note : Jouer vite et réduire ou augmenter la vitesse du galop par rapport à l'action qui se déroule à l'écran.

Tous droits réservés © Carl Fischer

Figure 1 : Exemple d'une *cue sheet* rédigée par Max Winkler

8 Richard Davis, *Complete Guide to Film Scoring: The Art and Business of Writing Music for Movies and TV*, Boston, Berklee Press, 1999, 2^{ème} éd. (2010), p. 24.

2. Les premiers films parlants : les débuts de la musique synchronisée

À la fin des années 1920, l'arrivée du son au cinéma révolutionne complètement la manière dont les films sont fabriqués, puisqu'en tant que vecteur de sens et de compréhension, le dialogue s'impose comme le premier point d'attention du spectateur. La musique de film et son système de production n'échappent pas à ce bouleversement : grâce à la synchronisation, un compositeur peut à présent s'immiscer dans la dramaturgie du film, et composer une pièce musicale qui accompagnera le film indépendamment de l'endroit où il sera projeté. Enfin, le passage du muet au parlant, c'est-à-dire de la musique interprétée dans la salle de projection au son synchronisé à l'image, est synonyme de chômage pour beaucoup de musiciens et de directeurs musicaux. Petit-à-petit, le statut et la fonction de la musique de film évoluent et passent du simple accompagnement au support de la dramaturgie et de la narration.

La voie vers le son synchronisé est montrée à New York en 1927 par la *Warner Bros.*, qui produit et distribue *Le Chanteur de Jazz*, dont le rôle principal est tenu par Al Jolson, un chanteur de variétés. Le film n'est pas complètement « parlant », mais contient quelques numéros musicaux dont le son est synchronisé à l'image. *Le Chanteur de Jazz* est un immense succès public et ouvre une nouvelle ère : celle du cinéma parlant ! Toutefois, à la fin des années 1920, il est encore coûteux et techniquement laborieux d'ajouter de la musique sur une scène de film en post-production, puisqu'il est encore impossible de l'enregistrer séparément : l'enregistrement multipistes n'est inventé qu'au milieu des années 1950. Autrement dit, les musiciens doivent être présents sur le plateau de tournage et jouer en même temps que les acteurs. A l'instar du *Chanteur de Jazz*, les premiers morceaux de musique synchronisés à l'image sont donc dans la majorité des cas des numéros musicaux, dans lesquels les musiciens sont même visibles à l'écran. Entre 1927 et 1931, les techniques d'enregistrement évoluent, et le début des années 1930 voit la naissance du « *dubbing* », que l'on peut traduire par

« doublage », mais qui correspond en réalité davantage au « mixage ». En effet, les Américains parlent indifféremment de « *dubbing* », de « *mixing* » ou de « *re-recording* » pour décrire l'opération de mixage, qui consiste, en post-production, à mixer des éléments sonores supplémentaires (effets, musique, etc.) avec le son direct enregistré sur le plateau de tournage, afin de réaliser la bande sonore du film.

Au début des années 1930, donc, la nécessité de justifier visuellement la musique qui accompagne une scène par la présence de musiciens à l'écran s'estompe, et les producteurs et les réalisateurs se dirigent vers deux extrêmes dans l'utilisation de la musique : si la musique n'est pas et n'a de toute façon plus besoin d'être justifiée à l'écran, il leur semble judicieux de placer de la musique sur toute la longueur du film, ou de ne pas en utiliser du tout. En parallèle, diverses expériences sont menées avec des morceaux pré-existants et à la « mode » du moment, dans un but de promotion des films. L'idée que la musique puisse entrer et sortir dans un film fait petit à petit son chemin, et l'usage de la musique de film, tel que nous le connaissons aujourd'hui, commence à se faire jour.

3. Le système des « studios » hollywoodiens

A la fin des années 1920, le nombre de films produits par Hollywood est exponentiel. Les studios, ou « *majors* », doivent donc s'organiser en conséquence et développent un système de production qui s'apparente à une chaîne d'assemblage industrielle. Chaque studio est une usine indépendante, qui possède tous les moyens de production d'un film et en contrôle toutes les étapes. Chaque studio possède également ses propres salles de cinéma, dans lesquelles sont projetés uniquement les films qu'il produit. Dans ce système, la fabrication d'un film est un travail d'équipe, dans le sens ouvrier du terme, et chacun a un rôle bien défini : si cette méthode peut sembler impersonnelle, il n'en demeure pas moins que de grands monuments de l'histoire du cinéma ont

été produits et réalisés durant cette période, comme par exemple *Tom Sawyer* (Paramount Pictures, 1930), *Scarface* (United Artists, 1932), *Heidi* (20th Century Fox, 1937), *The Wizard of Oz* (Metro-Goldwyn-Mayer, 1939), *Gone with the Wind* (Selznick International Pictures/Metro-Goldwyn-Mayer, 1939), et bien d'autres.

La musique n'échappe pas à ce système de production, et bien souvent, l'ensemble du département musical d'un studio est regroupé dans un même bâtiment. Ce dernier comprend en général une bibliothèque musicale, ainsi qu'un studio d'enregistrement. Tous les maillons de la chaîne de fabrication de la musique d'un film, des compositeurs aux musiciens, sont employés par le studio, et travaillent à demeure. Notons que durant cette période, toutes les personnes impliquées dans la fabrication d'un film, du réalisateur aux comédiens, en passant par les décorateurs, sont employés par un studio et travaillent uniquement pour des films produits par leur employeur. Chacun doit accomplir une tâche bien définie, et l'ensemble de la chaîne est compartimenté.

Le processus de fabrication de la musique d'un film d'alors est par bien des aspects similaire aux méthodes de production actuelles, à ceci près qu'aujourd'hui, il est d'usage qu'un seul compositeur soit responsable de la bande originale d'un film. De plus, tout le personnel employé par le producteur n'appartient plus à un studio : la fabrication d'un film est toujours un travail d'équipe, mais cette équipe est composée de personnes indépendantes, ou appartenant à de « petites » infrastructures, comme par exemple un studio d'enregistrement. Autrement dit, tout se fait aujourd'hui en-dehors du studio. Par contre, dans le modèle des studios qui s'étend de la fin des années 1920 aux années 1950, il n'est pas rare que plusieurs compositeurs travaillent sur un même film. Pour aller encore plus loin, bon nombre de compositeurs se « spécialisent » même dans un type de scène, et écrivent par exemple majoritairement la musique des scènes d'amour, ou des scènes de poursuite. En effet, les *majors* ont une certaine réticence à essayer de nouvelles choses,

tant au niveau de la fabrication du film qu'au niveau de la musique qui l'accompagne : Hollywood est alors très conservateur et préfère utiliser et réutiliser les recettes qui ont fait leurs preuves auprès du public plutôt que de prendre des risques qui pourraient représenter un péril économique.

Entre 1930 et 1950, environ cinq cents films par an sont produits par Hollywood, et le public américain n'a jamais autant fréquenté les salles de cinéma : on parle du *Golden Age*, ou de l'Âge d'Or d'Hollywood. À la fin des années 1930, huit studios hollywoodiens, les « *Big Five* » (*Metro-Goldwyn-Meyer (MGM)*, *Warner Brothers*, *20th Century Fox*, *Paramount* et *Keith-Albee-Orpheum (RKO)*) et les « *Little Three* » (*Universal*, *Columbia* et *United Artists*) se partagent 80 % de la production cinématographique nord-américaine.

Nous avons vu que la musique qui accompagnait les films muets avait un rôle complémentaire par rapport à l'ambiance d'une scène. A l'arrivée du cinéma parlant et du son synchronisé avec l'image, son rôle change drastiquement, puisque la musique doit à présent interagir avec le dialogue, tout en se développant en même temps que l'intrigue. C'est donc logiquement que les compositeurs d'origine européenne, qui ont de surcroît une culture et une pratique de l'opéra profondément enracinée, connaissent leur heure de gloire à Hollywood durant le *Golden Age*. Le style « Post-Romantique » d'Erich Korngold (1897-1957) ou de Max Steiner (1888-1971) est prédominant dans l'industrie cinématographique jusqu'à la fin des années 1950. Cependant, dès les années 1940, la première génération de compositeurs américains, comme Bernard Hermann (1911-1975) ou David Raksin (1912-2004), commence à étendre la palette des styles de la « musique hollywoodienne » en introduisant dans leurs partitions des éléments issus du Jazz ou de la musique contemporaine du XX^{ème} siècle. En 1941, alors que les méthodes de synchronisation de la musique à l'image n'ont qu'une dizaine d'années et que le style Post-Romantique de Korngold et Steiner règne en maître, le modèle établi est chamboulé tant sur le plan visuel que sur le plan musical par la sortie de *Citizen Kane*, un film réalisé par Orson Welles et mis en musique par

Bernard Hermann. Après ce bouleversement stylistique et créatif, l'arrivée de la télévision une dizaine d'années plus tard va à nouveau modifier durablement l'économie de l'industrie cinématographique.

4. L'arrivée de la télévision et la fin du système des studios

Une vingtaine d'années après l'avènement du cinéma parlant, c'est-à-dire à la fin des années 1940, la télévision fait son apparition sur le marché destiné au grand-public. Dans un premier temps, la plupart des grands studios hollywoodiens espèrent qu'elle ne constitue qu'une passade, qu'un effet de mode, et que la télévision ne s'implantera pas durablement. En effet, au cœur du *Golden Age*, l'industrie hollywoodienne produit des centaines de films et génère des centaines de millions de dollars de recette par an grâce aux millions de spectateurs qui se rendent chaque semaine dans les salles obscures. La tourmente déclenchée par l'arrivée de la télévision perdure, et entre 1955 et 1970, le système des studios chavire. Selon le rapport de l'INHESJ., Hollywood redéfinit alors le long-métrage comme un « produit haut de gamme » et développe des formats larges, plus spectaculaires, comme le *CinémaScope* ou le *VistaVision*. En parallèle, les *majors* créent des filiales destinées à produire pour la télévision, considérant que la télévision constitue un formidable débouché pour d'autres catégories de longs-métrages et d'autres types de programmes⁹.

Si l'arrivée de la télévision et sa commercialisation à grande échelle constituent un facteur majeur du chamboulement du système des studios, puisqu'elles induisent une baisse significative de la fréquentation des salles de cinéma, il ne faut pas non plus négliger l'effet du procès anti-monopole d'Hollywood de 1948. Durant ce procès, aussi appelé « décision *Paramount* », la Cour suprême des États-Unis, en vertu du droit à la concurrence, décide que les studios ne peuvent plus posséder leurs propres chaînes de salles de

9 Groupe de veille et d'analyse « MALRAUX », *op.cit.*, p. 8.

cinéma. Dès 1949, les studios hollywoodiens sont donc contraints de se séparer de leurs parcs de salles, et de cesser leurs pratiques monopolistiques ; c'est une désintégration verticale de l'industrie cinématographique nord-américaine¹⁰, qui jusqu'ici maîtrisait l'ensemble de la chaîne cinématographique, de la conception à l'exploitation. Il faudra attendre l'élection de Ronald Reagan au début des années 1980 pour que les *majors* trouvent un accord avec l'État américain, et reconstruisent leur monopole en dépit de la loi *antitrust* toujours en vigueur¹¹.

Enfin, le maccarthysme, ou la « chasse aux sorcières », qui sévit entre 1950 et 1954 sur le territoire américain affaiblit économiquement Hollywood. Un certain nombre d'employés des studios, à tous les niveaux de la chaîne de production et de la hiérarchie, ne peuvent plus travailler : une liste noire d'artistes et de techniciens, à qui les studios refusent tout emploi, est même créée.

La cumulation de ces trois facteurs conduit irrémédiablement à une baisse significative des recettes des studios, qui se voient dans l'obligation de licencier une grande partie de leur personnel. Pour autant, cela ne signifie pas l'arrêt de la production cinématographique, mais plutôt l'émancipation et la libéralisation des différents secteurs sollicités pour la fabrication d'un film. En d'autres termes, les studios ne peuvent plus assumer économiquement l'ensemble des moyens de production d'un film et, en l'espace de quelques années, la dynamique de production cinématographique nord-américaine change complètement.

Nous l'aurons compris, les studios ne ferment pas leurs portes, mais leur rôle économique et créatif évolue, pour se rapprocher de celui que nous connaissons aujourd'hui. Aux États-Unis d'Amérique, les studios ont désormais pour rôle de fournir et/ou de réunir les fonds nécessaires à la production d'un film, de son tournage à sa distribution : les studios sont désormais des

10 Groupe de veille et d'analyse « MALRAUX », *op. cit.*, pp. 7-8.

11 *ibid.*, pp. 8-9.

« banques spécialisées », qui investissent dans des projets conçus par d'autres et servent d'intermédiaires¹². Ils ne maîtrisent plus tous les maillons qui composent la chaîne de fabrication d'un film, bien qu'ils contrôlent encore dans la majorité des cas la validation ou non du « produit final », avant sa distribution et son exploitation. Ainsi, les producteurs exécutifs¹³, mais aussi les comédiens, les techniciens, et tous les métiers créatifs dont les compositeurs, deviennent progressivement indépendants. Ils peuvent alors travailler pour plusieurs studios, au gré des projets pour lesquels ils sont engagés. Pour résumer, la chaîne de production d'un film est restée relativement similaire à celle qui avait été établie en termes de processus créatifs et techniques, mais elle a quitté les infrastructures et le contrôle total des studios pour se répartir entre de nombreux maillons indépendants.

5. De la fin des studios à aujourd'hui : la révolution numérique

Comme nous venons de le voir, l'arrivée de la télévision à la fin des années 1940 a eu des répercussions économiques irréversibles sur l'industrie cinématographique. Elle a aussi eu une influence stylistique et esthétique non négligeable, puisqu'une fois qu'elle a été implantée dans bon nombre de foyers, les producteurs, qui ont toujours été attentifs aux goûts du public, se sont beaucoup inspirés des programmes télévisuels.

Durant le *Golden Age*, le style musical prédominant à Hollywood est celui des larges orchestres post-romantiques, ce qui n'empêche pas le public

12 Groupe de veille et d'analyse « MALRAUX », *op. cit.*, p. 8.

13 Dans l'industrie cinématographique, les producteurs exécutifs sont responsables du financement du film et/ou de l'effort créatif nécessaire à son aboutissement, mais ils n'interviennent pas sur le plateau. Les responsabilités du producteur exécutif peuvent donc comprendre le financement de tout ou partie du film et/ou la recherche d'investisseurs — ici, les studios —, mais aussi la résolution de questions légales, la rédaction du scénario, la publicité et la promotion du film ; pour résumer, les producteurs exécutifs ont un rôle de conseil et de supervision pour l'ensemble de la fabrication du film qu'ils produisent.

d'écouter aussi des artistes jouant de la musique pop/rock ou jazz. Dans les années 1960 et 1970, tandis que *The Rolling Stones* ou Frank Sinatra vendent des millions de disques, la musique de film s'ouvre stylistiquement à ces influences. A cette époque, les thèmes des programmes télévisuels sont plutôt « rock » ou « jazzy » et les producteurs de cinéma sont très influencés par la musique composée pour la télévision. Cette mutation et cette ouverture stylistiques de la musique de film préparent l'avènement des partitions « orchestrales/pop », ou « pop/orchestrales », qui domineront les années 1980 et 1990.

Toutefois, au milieu des années 1970, tandis que la musique orchestrale pour le grand écran est en perte de vitesse, Steven Spielberg réalise *Jaws* (« *Les Dents de la mer* », 1974), accompagné par une partition orchestrale de John Williams. D'après Richard Davis¹⁴, cette partition de Williams marque la résurgence du style orchestral du XIX^{ème} siècle, ou plus précisément d'un vocabulaire musical Néo/Post-Romantique, que l'on retrouve quelques années plus tard dans la trilogie originale de la saga écrite et en partie réalisée par Georges Lucas, *Star Wars*¹⁵. Le style de Williams et le vocabulaire musical qu'il remet au goût de l'époque, se caractérisent notamment par des thèmes lyriques, des *tutti* de cuivres triomphants, une fine écriture des bois, etc. ; il ne faut pas oublier que ces nouvelles partitions orchestrales n'hésitent pas à utiliser des techniques de composition et d'orchestration contemporaines quand elles s'avèrent nécessaires et appropriées. Si le genre de la musique de film orchestrale se renouvelle, le début des années 1980 va être marqué par l'arrivée de toute une palette de sons et de nouveaux instruments électroniques : le monde de la composition musicale va à nouveau être chamboulé par l'arrivée des synthétiseurs d'une part, et de l'informatique [musicale] d'autre part.

14 Richard Davis, *op. cit.*, pp. 59-60.

15 *Star Wars: Episode IV – A New Hope / La Guerre des Étoiles : Épisode IV – Un Nouvel Espoir* (1977) ; *Star Wars: Episode V – The Empire Strikes Back / La Guerre des Étoiles : Épisode V – L'Empire contre-attaque* (1980) ; *Star Wars: Episode VI – Return of the Jedi / La Guerre des Étoiles : Épisode VI – Le Retour du Jedi* (1983).

En 1981, la musique composée par Vangelis pour le film *Chariots of Fire* (« *Les Chariots de feu*») est la première bande originale, composée presque uniquement avec des synthétiseurs, à remporter un Oscar. Nous développerons plus en détail les évolutions méthodologiques et techniques engendrées par cette révolution technologique, mais nous retiendrons pour le moment que le monde de la musique dans son ensemble est définitivement modifié par l'arrivée des synthétiseurs, puis de l'informatique musicale. Au milieu des années 1980, les synthétiseurs sont de plus en plus accessibles et abordables, et toutes les générations de compositeurs de musique [à l'image] les utilisent. Dès la fin des années 1970, Hans Zimmer et son équipe s'intéressent en profondeur à ces nouvelles technologies, et à la création de sons inédits par le biais d'échantillonneurs et de synthétiseurs analogiques ou numériques. L'arrivée des sons « électroniques », de l'ordinateur et de l'informatique musicale, constituent une révolution à la fois technique, technologique, esthétique et méthodologique, notamment dans le monde de la musique de films.

B. Traditions et évolutions des pratiques de composition de musique à l'image

Selon Richard Davis, « le monde de la musique de film est centré autour de l'usage de la technologie »¹⁶. Nous avons déjà pu nous en rendre compte au-travers de son histoire, mais nous allons voir que les méthodes de travail des compositeurs de musique à l'image sont étroitement liées à l'évolution des outils et des technologies mis à leur disposition. Il nous semble donc judicieux de nous attarder sur ces méthodes de travail, sur leurs évolutions et sur l'influence que les différentes technologies développées depuis le début du XX^{ème} siècle ont eues sur ces dernières, afin de mieux comprendre la chaîne et le flux de travail que nous connaissons aujourd'hui.

1. Les premières méthodes de travail

Durant près de cinquante ans, des années 1930 aux années 1980, l'enregistrement analogique est le standard de l'enregistrement sonore et musical ; il est d'ailleurs uniquement monophonique jusqu'au milieu des années 1960. Jusqu'à la fin des années 1950, c'est-à-dire jusqu'à la possibilité d'enregistrer en multipistes et de synchroniser la musique à l'image, la musique et le « son » du film doivent être enregistrés ensemble et simultanément. Précisons ici que bien qu'il soit une option depuis la fin des années 1950, l'enregistrement multipistes n'a pas été très utilisé pour la musique de film orchestrale, puisqu'il est beaucoup plus efficace et « musical » d'enregistrer l'orchestre dans son ensemble, d'autant plus que le montage sur bande magnétique, puis sur DAT¹⁷, pouvait s'avérer long,

16 « The film-scoring world is centered around the use of technology. » — Richard Davis, *op. cit.*, p. 68.

17 Le DAT (« *Digital Audio Tape* ») est un format de cassette audio, qui permet d'enregistrer et de lire des données numériques sur une bande magnétique de 3,81 mm. Ce format a été conçu par *Sony* à la fin des années 1980.

compliqué et fastidieux ; aujourd’hui, tout est enregistré grâce à un logiciel dédié — *ProTools* (*Avid*) dans l’écrasante majorité des cas, et dans l’industrie cinématographique en particulier —, accessible et édité depuis un disque dur, interne ou externe.

Durant toute cette période, les compositeurs travaillent sur papier, et composent sous la forme d’ébauches de quelques portées musicales, qui sont ensuite orchestrées par eux-mêmes ou par leurs collaborateurs. La copie des partitions, c’est-à-dire du conducteur, et des parties séparées¹⁸, est également réalisée manuellement. Il faut attendre le début des années 1990 pour que des logiciels de notation et de gravure musicales suffisamment aboutis voient le jour, et une dizaine d’années supplémentaires pour que ces solutions logicielles soient généralisées et implantées définitivement dans l’industrie. A titre d’exemple, la première version de *Finale* (*MakeMusic, Inc.*) est commercialisée en 1988, et celle de *Sibelius* (*Avid*) en 1993. Ces deux logiciels se partagent aujourd’hui la majorité du marché, bien que *Dorico*, développé par *Steinberg* depuis 2016, semble vouloir perturber cet équilibre établi depuis une vingtaine d’années.

2. L’arrivée des synthétiseurs et de l’informatique

Nous l’avons déjà évoqué : l’arrivée des synthétiseurs et de nouveaux sons électroniques jusqu’alors inexistantes induit un premier bouleversement du paysage musical de la fin des années 1970. Les premiers synthétiseurs utilisés dans la musique de film sont des instruments analogiques, comme le thérémine, le *ARP 2500* (1970), ou le premier modèle de *Moog* (1964). Ces

18 Le « conducteur » est une partition de large format sur laquelle sont notées toutes les parties jouées par tous les instruments de l’orchestre, ou plus largement de l’ensemble instrumental pour lequel la pièce musicale a été écrite ; le *conducteur* est destiné au chef d’orchestre. Les « parties séparées », *a contrario*, sont destinées aux musiciens, et ne présentent que la ou les partie(s) jouée(s) par un instrumentiste. On peut parfois trouver plusieurs parties sur une même *partie séparée*, notamment pour les percussions.

instruments, notamment les synthétiseurs ARP et Moog, sont relativement complexes, et peu de compositeurs sont alors en mesure de les utiliser : il n'est donc pas rare qu'ils aient recours à des « experts » uniquement chargés de programmer ces instruments. Assez rapidement, les constructeurs de synthétiseurs développent des instruments plus « faciles » à utiliser : en 1978, la société *Sequential Circuits* commercialise le *Prophet-5*, et trois ans plus tard, en 1981, c'est au tour de *Roland* de sortir le *Jupiter-8* sur le marché relativement grand-public des synthétiseurs analogiques. L'année 1983 marque un tournant décisif pour l'informatique musicale : alors que *Yamaha* lance sur le marché le premier synthétiseur numérique polyphonique, le *DX7*, la norme MIDI est déjà en cours de développement. Cette norme permet d'une part de traduire des notes musicales — et de plus en plus d'autres paramètres — en informations digitales, et d'autre part de faire communiquer plusieurs machines — au départ, des synthétiseurs — entre elles. En d'autres termes, la norme MIDI est dans un premier temps pensée comme un protocole de communication universel entre toutes sortes de machines musicales. Nous consacrerons toute une partie à cette norme, qui comme nous le verrons, est aujourd'hui au cœur des outils et des méthodes de travail des compositeurs.

En parallèle du déploiement des synthétiseurs sur le marché, la Musique Assistée par Ordinateur (MAO) se développe au début des années 1990. À titre d'exemple, la société allemande *Steinberg* commercialise au mois d'avril de l'année 1989 *Cubase 1.0 Atari*, qui est un séquenceur capable de gérer uniquement le MIDI. En 1991, *Cubase Audio*, disponible uniquement sur *Macintosh*, permet aussi d'éditer des sons, et en 1992, *Cubase* est porté sur *Windows*. Il n'a depuis cessé d'être développé pour devenir la Station de Travail Audio-Numérique (STAN)¹⁹ que nous connaissons aujourd'hui. On retrouve la même chronologie pour *Logic*, qui est développé à la fin des années 1980 par Gerhard Lengeling et Chris Adam sous la forme d'un séquenceur

19 En anglais, nous parlerons de DAW, pour « *Digital Audio Workstation* ». Nous détaillerons les spécificités et les fonctionnalités d'une STAN un peu plus tard, mais nous retiendrons pour l'instant qu'il s'agit tout simplement d'un logiciel de MAO.

MIDI, toujours sur *Atari* ; il porte alors le nom de *Creator*. En 1993, la société *Emagic* commercialise une nouvelle version de *Logic* sous le nom de *Notator Logic*, toujours sur *Atari*. Assez rapidement, le logiciel est porté sur les systèmes d'exploitation *Macintosh* et *Windows*, et en juillet 2002, *Apple* rachète *Emagic*, et arrête rapidement de développer la version *Windows* du logiciel. *Logic Pro X* (10.4) est à ce jour l'une des STAN les plus complètes et les plus utilisées dans le monde de la musique de film. Enfin, il nous faut citer le cas de *Digital Performer*, développé par *Mark of the Unicorn* (MOTU). Il s'agit d'une STAN principalement dédiée à la composition de musique à l'image de par les fonctionnalités qu'elle propose ; toutefois, *Digital Performer* est surtout répandu chez les compositeurs américains. En 1984, MOTU développe *Professional Composer*, uniquement pour *Macintosh*, qui permet de réaliser des partitions sommaires. L'année suivante, la société commercialise *Performer*, un séquenceur MIDI, et en 1990, MOTU ajoute la possibilité de synchroniser des fichiers audio dans *Performer*, et combine *Professional Composer* et *Performer*, pour donner naissance à *Digital Performer*.

Ce dernier exemple montre clairement que les développeurs de logiciels dédiés à la musique ont d'abord travaillé sur le MIDI dans la deuxième moitié des années 1980, pour répondre à la demande croissante des utilisateurs de synthétiseurs. En parallèle, les logiciels plutôt dédiés au traitement de l'audio se sont aussi développés, comme *ProTools* (1989) ou *MAGIX Samplitude* (1992), puis progressivement, chacun s'est ouvert et a élargi sa palette de fonctionnalités pour donner naissance aux STAN que nous connaissons aujourd'hui, et qui font cohabiter le MIDI et l'audio. C'est du moins le cas des STAN utilisées pour la composition et la production de musique de film.

3. Les outils de travail actuels

Si Richard Davis fait le constat que le monde de la musique de film est centré autour de la technologie, il ajoute que « le compositeur moderne de

musique de film est entouré d'ordinateurs contenant des logiciels onéreux, divers appareils externes, et souvent d'un ou deux assistant(s) pour l'aider avec la programmation MIDI et d'éventuels problèmes techniques »²⁰. Il ajoute que « l'usage de la technologie a par bien des manières aidé le processus de composition de musique à l'image. [La technologie] peut aider à communiquer des idées musicales aux réalisateurs *via* une maquette d'une manière qu'un compositeur jouant au piano et fredonnant une mélodie ne pourrait jamais atteindre »²¹. Aujourd'hui, nous sommes capables de reproduire fidèlement le son de la plupart des instruments acoustiques, des synthétiseurs mythiques des années 1980, ou encore créer des sons inédits grâce à l'informatique musicale. Il est possible de créer à la volée, grâce à un clavier maître²² des maquettes qui sonnent « comme » un orchestre symphonique, ou tout autre ensemble instrumental. De cette manière, le compositeur peut faire écouter au réalisateur ou aux producteurs d'un film des maquettes vraisemblables, et ressemblantes à la musique qui sera ensuite enregistrée. Auparavant, le réalisateur découvrait ce qui allait devenir la musique de son film au moment de la session d'enregistrement, puisque le compositeur n'avait pas d'autre options que de lui jouer au piano et de lui fredonner chaque morceau, en lui expliquant comment il serait orchestré ; nous discuterons des tenants et des aboutissants de ces méthodes de composition et de production plus tard.

La technologie a également permis à des compositeurs très talentueux mais sans formation musicale, ou du moins avec de faibles connaissances théoriques mais une pratique musicale empirique, de travailler, et d'intégrer le

20 « The modern film composer is surrounded by computers containing thousands of dollars' worth of software, various outboard devices, and often an assistant or two to help with MIDI programming and technical issues. » — Richard Davis, *op. cit.*, p. 68.

21 « [...] the use of technology has in many ways aided the film-scoring process. It can help to communicate musical ideas to directors *via* a mock-up in a way that the composer playing the piano and humming a melody never could. » — Richard Davis, *op. cit.*, p. 76.

22 Un *clavier maître* est un clavier de piano qui permet de piloter en MIDI notre STAN et divers instruments virtuels. En d'autres termes, il est possible de « jouer » du violon, du saxophone ou encore de la batterie grâce à un clavier maître.

milieu de la musique de film. Il y a bien sûr des mauvais côtés inhérents à l'omniprésence de la technologie : tout d'abord, elle évolue [très] vite, et il est compliqué de constamment maintenir à jour ses connaissances, ses compétences, et son matériel. Ensuite, utiliser des échantillons, c'est-à-dire des instruments virtuels, et pouvoir entendre instantanément le résultat quasi-définitif de ce que l'on est entrain d'écrire peut inhiber l'imagination, ou encore se substituer à l'apprentissage du fonctionnement de chaque instrument : c'est notamment pour cette raison que le rôle de l'orchestrateur a de plus en plus de poids dans l'élaboration de la bande originale d'un film.

Pour conclure, la technologie doit être utilisée comme un outil et un moyen, c'est-à-dire comme l'une des nombreuses dimensions de l'ensemble des compétences que doit acquérir et maîtriser un compositeur, au même titre que sa connaissance de l'harmonie, de l'orchestration, ou encore de l'organologie.

C. La fabrication d'une musique de film

Avant de nous intéresser spécifiquement à la chaîne de fabrication d'une musique de film, il nous faut la replacer dans son contexte, c'est-à-dire décrire succinctement la chaîne de fabrication d'un film. Cette mise en perspective nous permettra de mieux comprendre la place créative et technique de la musique dans la fabrication d'un film, et d'en déduire les contraintes inhérentes au fait qu'elle s'inscrive dans la continuité et la finalité d'un processus plus macroscopique.

1. Rappels : la chaîne de fabrication d'un film

Dans le processus de fabrication d'un film, nous comptons deux entités décisionnelles principales : la production, qui prend en charge les aspects financiers et organisationnels du processus, et la réalisation, responsable de la partie créative. Pour la concrétisation et l'aboutissement d'un projet cinématographique, il est plus que souhaitable que ces deux entités s'entendent et se comprennent parfaitement, mais dans la majorité des cas, la production doit faire face à un certain nombre de contraintes, qui peuvent la faire rentrer en conflit avec la réalisation, bien que chacun souhaite aboutir au meilleur résultat possible. Ainsi, qu'ils soient techniques ou créatifs, tous les partenaires doivent répondre à la fois à la production et à la réalisation, ce qui est parfois un équilibre complexe. Quoiqu'il en soit, chacun a un ou des rôle(s) bien précis, qu'il doit accomplir à certaines étapes de la fabrication du film.

Ces étapes se répartissent sur trois grandes phases, qui sont la *pré-production*, la *production*, et la *post-production*. Avant de décrire sommairement ces trois étapes, n'oublions pas de garder à l'esprit que l'industrie cinématographique doit être flexible. En effet, comme le dit le réalisateur Jean-Pierre Melville : « le métier du cinéma n'est comparable à aucun autre. [...] Chaque film est un prototype. Une pièce de théâtre cesse

d'en être un si elle dépasse la centième. L'effort de chaque représentation n'est pas fait en vain. Tandis que, pour toujours, l'effort de création, de tournage, de distribution et d'exploitation d'un film demeurera un risque total. C'est le métier le plus dangereux du monde. »²³. Si les derniers propos de Melville sur le « métier du cinéma » doivent sans aucun doute être nuancés, nous retiendrons que chaque film est un prototype, tant sur le plan de sa conception, de sa fabrication, que de sa finalité. C'est pour cette raison que nous donnerons ici un aperçu de la chaîne de fabrication « type » d'un film, qui pourra, et devra varier suivant un grand nombre de paramètres, notamment budgétaires et organisationnels. Autrement dit, il n'existe pas de chaîne de production universelle, et la chaîne théorique que nous rappelons ici doit être considérée comme un « patron », un modèle auquel chaque film va essayer de se rapprocher sans pour autant en être le clone ; chaque film est un cas particulier de ce modèle économique et productif.

1.1. La pré-production

La *pré-production* constitue la première étape dans le processus de création d'un film. Elle peut être initiée par la production, la réalisation, un scénariste, ou diverses combinaisons de ces différentes entités, et dans de multiples circonstances. A vrai dire, il n'y a pas de règle : un scénariste peut proposer un *pitch*²⁴ à un producteur, un réalisateur peut contacter un producteur avec une continuité dialoguée, ou encore un producteur peut avoir acquis les droits d'un roman et en commander l'adaptation, etc. Dans tous les cas, une équipe restreinte élabore une histoire, qui sera concrétisée sous la forme d'un scénario.

23 Jean-Pierre Melville dans un entretien avec Michel Mardore, *Candide* n°205, 29 mars 1965.

24 Dans le cinéma, et plus largement pour une œuvre de fiction, un *pitch* est une synthèse de l'histoire qui est racontée en une phrase, ou un petit paragraphe.

Une fois que le scénario est achevé, ou du moins assez avancé, la production se lance à la recherche de financements pour produire le film, engage les principaux collaborateurs de création, organise le recrutement des actrices et des acteurs qui endosseront le rôle des différents personnages du film, c'est-à-dire le « *casting* », organise et prépare le calendrier du tournage, recherche des décors naturels ou des studios suivant les besoins du film, et enfin, engage l'équipe technique du film. Pour résumer, la pré-production comprend toutes les étapes nécessaires à l'écriture de l'histoire, au financement et à l'organisation du tournage du film : elle s'étend donc théoriquement de l'idée du film, soit de l'élaboration de l'intrigue, jusqu'au premier jour de tournage.

1.2. La production

La *production* désigne tout simplement le tournage du film. Cette étape comprend les répétitions avec les actrices et les acteurs, le tournage en studio et/ou dans des décors extérieurs, le visionnage des *rushes*²⁵ au jour le jour, ainsi que le début du travail sur les effets spéciaux visuels, lequel continuera et s'achèvera en post-production. En général, les monteurs de l'image du film commencent à trier et à organiser les *rushes* tournés, et parfois, débudent même le montage de certaines séquences.

1.3. La post-production

La *post-production* intervient après le tournage, et peut être partagée en deux départements, bien que les technologies numériques aient tendance à

25 Au cinéma, les *rushes*, ou « *épreuves de tournage* », désignent l'ensemble des documents originaux, qu'ils s'agissent d'images ou de sons, produits au tournage. C'est pour cette raison que le tournage correspond à l'étape de « production », puisque c'est à ce moment-là que la matière originale qui permettra de fabriquer le film est créée.

réduire les frontières entre ces derniers : le département « Image », et le département « Son ». C'est en général à cette étape que le travail sur la musique commence. Il est tout à fait possible, voire recommandé, que le compositeur, la réalisation et la production aient commencés à travailler en amont sur des idées musicales, mais le travail de composition « à l'image » débute vraiment après le montage de l'image du film. C'est à partir de ce moment-là que la chaîne de production de la musique du film se met en branle, au fur et à mesure que le compositeur avance dans son travail de composition.

Le département Image s'occupe de monter l'image du film, et le cas échéant de créer et de réaliser des effets spéciaux visuels. Il est aussi en charge de l'étalonnage, c'est-à-dire des corrections colorimétriques qui doivent être apportées pour des raisons techniques, esthétiques, et narratives. Quoiqu'il en soit, il ne nous semble pas pertinent de développer ici davantage les activités de ce département en post-production : nous retiendrons simplement qu'il s'occupe de livrer le montage définitif du film au compositeur.

Le département Son, quant à lui, peut se diviser en quatre sous-départements principaux : le montage son, le bruitage, le mixage, et la musique. Le montage son consiste à éditer les *rushes* par rapport au montage image, à monter et créer des effets sonores, des ambiances, etc. Le cas échéant, il peut être nécessaire de réenregistrer certaines répliques en post-synchronisation pour des raisons techniques, de jeu, ou encore de changement(s) de texte : cette séance d'enregistrement est en général placée sous la direction du mixeur du film. Le bruitage consiste à recréer des sons grâce à divers objets, qui en général sont totalement décorrélés de la source originale du son reproduit — des noix de coco pour reproduire le pas d'un cheval, par exemple. Le bruitage permet de renforcer la présence des personnages à l'écran, de créer des effets sonores originaux, ou de remplacer des éléments sonores qui seront perdus lors de la fabrication de la version

internationale (V.I.)²⁶ du film. Le mixage, enfin — nous n’oublions pas le département musique, qui sera abordé beaucoup plus longuement ensuite —, constitue l’ultime étape de fabrication de la bande sonore du film : il consiste à mélanger et à spatialiser l’ensemble des sources qui constituent la bande sonore du film, à savoir le son direct enregistré sur le tournage, c’est-à-dire les *rushes*, avec les dialogues éventuellement post-synchronisés, les effets sonores, les ambiances, les bruitages, et la musique.

Une fois que l’étalonnage d’une part, et le mixage d’autre part sont achevés, il reste à fabriquer tous les éléments nécessaires à la distribution du film : le DCP²⁷ pour les salles de cinéma, ainsi que les exports nécessaires au pressage de DVD, Blu-Ray, ou à la diffusion du film sur des plates-formes de vidéo à la demande, etc.

Comme nous allons le voir, la fabrication de la musique du film se fait dans sa majeure partie en parallèle de toutes ces étapes de post-production. Outre les contraintes budgétaires, le compositeur et son équipe doivent surtout faire face à d’importantes contraintes de temps, puisque la musique est le « tampon » entre le montage de l’image du film et le mixage de sa bande sonore. En effet, le compositeur travaille autant que faire se peut sur un montage définitif ; il doit donc attendre que ce dernier soit terminé, mais il doit

26 La version internationale (V.I.) d’un film est une copie de la version originale (V.O.) qui servira de support à la fabrication de toutes les version étrangères du film. Pour fabriquer une V.I., on enlève les dialogues dans la langue source, ainsi que les titres. Pour compenser les éléments sonores perdus avec la suppression des dialogues, c’est-à-dire du son direct, on doit ajouter des bruitages supplémentaires, qu’on ne retrouvera pas dans la V.O. du film. Pour chaque version étrangère, on ajoute donc à la V.I. des bruitages, mais aussi des effets et des ambiances sonores propres à la langue cible, et enfin des titres, sous-titres, génériques de début et de fin dans la langue cible. Des comédiens doublent ensuite chaque dialogue dans la langue cible lors d’une séance de post-synchronisation ; c’est le « doublage » du film.

27 Pour schématiser, le *Digital Cinema Package* (DCP) est l’équivalent en cinéma numérique de la copie de projection argentique. C’est donc le DCP qui sera distribué aux exploitants des salles de cinéma et qui sera projeté au public.

aussi livrer sa musique enregistrée et prémixée pour le mixage, dont la date de début est souvent fixée dès la pré-production, tout comme la date de sortie du film. Il n'est pas rare que les dates de la session d'enregistrement soient également arrêtées assez tôt dans le processus. En d'autres termes, le compositeur et son équipe sont tributaires de l'éventuel retard pris en production et en début de post-production, sans pour autant pouvoir allonger le temps de travail qui leur est initialement alloué.

2. Composition de la musique d'un film : terminologie et travaux préliminaires

Le recrutement et le début du travail du compositeur sur un film peut s'opérer de plusieurs manières et à divers moments. Tout cela dépend de nombreux facteurs humains, calendaires et budgétaires : certains compositeurs aiment commencer à travailler dès l'écriture du scénario, d'autres avec les premiers *rushes*, ou encore une fois que le montage est terminé ; certains préfèrent même travailler sans image. Cela dépend aussi de la relation et des antécédents qui existent ou non entre le compositeur et le réalisateur, ainsi que des méthodes de travail de chacun. Quoiqu'il en soit, ces premières étapes de recherche ne nous concernent pas au premier chef, et doivent demeurer le privilège de ceux qui s'y confrontent. Avant de rentrer dans le vif des travaux préliminaires qui marquent le début effectif du travail du compositeur et de son équipe, il nous semble nécessaire de préciser ici quelques éléments pratiques et terminologiques.

En général, un long-métrage de fiction standard comporte entre trente minutes et deux heures de musique. Notons ici qu'un compositeur expérimenté comme John Williams considère qu'il a accompli une bonne journée de travail lorsqu'il a composé deux minutes de musique. Toutefois, selon Tristan Noon²⁸,

28 Tristan Noon, *From DAW to Score: An Essential Guide to Music in Film and TV, and Transforming Your MIDI Data into a Score Ready for Recording Sessions*, Londres, Tristan Noon Music, 2017, p. 11.

une production télévisuelle attend d'un compositeur qu'il compose la musique d'un épisode entre une et deux semaine(s), tandis que les temps de travail dans l'industrie cinématographique sont [beaucoup] plus longs, et peuvent varier entre quatre semaines et six mois.

Dans un film, et plus largement dans tout programme audiovisuel, chaque morceau de musique est appelé un « *cue* », tandis que l'ensemble des *cues* est désigné par le « *score* », c'est-à-dire la « partition » du film. Aujourd'hui, ces anglicismes sont entrés dans le vocabulaire des compositeurs francophones, qui emploient quotidiennement les termes « *cue* » et « *score* », ce que nous ferons donc également. La dénomination d'un *cue* est codifiée, bien que cette codification puisse présenter des variations suivant le contexte dans lequel elle est utilisée. Généralement, les *cues* sont nommés ainsi : « 6M08 », soit un premier nombre, la lettre « M », puis un second nombre. Dans le cinéma, cela signifie qu'il s'agit du huitième *cue* de la sixième bobine²⁹ du film. À la télévision, on ne travaille pas en bobines, mais en « épisodes », ou en « actes ». Les actes désignent les différents segments d'un programme séparés par les annonces publicitaires. Pour une série, le premier nombre du nom d'un *cue* correspond donc soit à l'épisode d'une saison donnée, soit à un acte ; un épisode est composé de plusieurs actes, mais leur compte n'est pas réinitialisé à chaque épisode, pour des raisons évidentes de redondance de dénomination des fichiers pour une même saison. La lettre « M » est tout simplement l'abréviation du terme « *Music* ». Bien évidemment, il existe des variantes de cette nomenclature : par exemple, certains compositeurs ajoutent la version du *cue*, ce qui nous donnera ici, pour la troisième version de notre *cue* : « 6M08v3 ». Ce principe de dénomination des *cues* est très largement utilisé par les compositeurs qui évoluent dans l'industrie cinématographique occidentale, et sans doute mondiale.

29 Avant la numérisation de l'industrie cinématographique, un film était découpé en plusieurs bobines de six cents mètres, équivalentes à une durée de projection d'environ vingt minutes. Aujourd'hui, il est encore d'usage de travailler « en bobines », c'est-à-dire de diviser le film en séquences d'environ vingt minutes, qui sont ensuite assemblées pour assurer la continuité et l'unité du film.

Comme nous l'avons évoqué, il est possible qu'un compositeur commence à travailler sur la musique d'un film avant même que le tournage n'ait commencé. Il peut proposer des idées musicales au réalisateur et au(x) producteur(s), mais le travail à proprement parler sur la bande originale d'un film commence une fois que le montage de l'image est achevé ou en cours de finalisation. C'est à partir de ce moment-là que tous les collaborateurs du compositeur sont sollicités et que celui-ci consacre l'intégralité de son temps de travail au projet en question.

Le processus commence par une « réunion de repérage », ou « *spotting session* » : elle marque le point de départ de la création musicale à l'image, puisque c'est à cette occasion que le compositeur, la réalisation et la production décident des endroits où placer la musique dans le film, mais aussi quelles couleurs musicales et instrumentales employer, et enfin, quel(s) rôle(s) narratif(s) doit endosser la musique. C'est à cette rencontre que s'élabore théoriquement la musique du film. Aussi, il est décisif d'établir durant cette réunion de repérage une communication claire et franche entre le compositeur et le réalisateur, afin d'échafauder des fondations communes pour bâtir ensuite le *score*. En d'autres termes, c'est à ce moment-là qu'une grande partie des décisions importantes relatives à la bande originale du film sont prises.

Trois collaborateurs, ou trois groupes de collaborateurs, accompagnent le compositeur au jour le jour dans son travail de composition : son assistant, l'orchestrateur, et le monteur de la musique. Ce dernier est essentiel dans l'élaboration de la bande originale d'un film, bien que ce poste ne soit pas systématiquement pourvu en France, et qu'il ne revête de toute façon pas l'importance qui le distingue dans la chaîne de fabrication de la musique à Hollywood, où le « *music editor* » est un poste-clé. En effet, le travail du monteur de la musique du film commence en général avant celui du compositeur.

Au milieu de la post-production, alors que le montage de l'image est en passe d'être achevé, le monteur de la musique est chargé de monter sur

l'image du film des pistes de « musique temporaire », ce que l'on appelle la « *temp music* » ou la « *temp track* ». Cela consiste à monter des morceaux de musique préexistants sur l'image du film, afin de donner une première idée des envies du réalisateur au compositeur, mais aussi de travailler plus aisément sur le rythme du montage de l'image et éventuellement de montrer le film à des spectateurs « tests ». La musique temporaire donne une première couleur musicale à chaque scène, mais chaque morceau de musique temporaire sera remplacé par la musique composée pour le film. Paula Flach rappelle que contrairement à d'autres étapes de la fabrication d'une musique de film, « les pistes de musique temporaire constituent un développement relativement récent. Aux débuts du cinéma, par exemple durant l'Âge d'Or d'Hollywood, il n'y avait pas de musique temporaire, puisque cette pratique était technologiquement irréalisable »³⁰. Celle-ci est d'ailleurs souvent décriée par les compositeurs, car elle aurait tendance à inhiber leur créativité et à les cantonner à produire une imitation de la musique temporaire, mais cela ne l'a pas empêchée de se généraliser. L'exemple le plus célèbre est sans doute celui de *Star Wars*, qui a notamment été monté sur des extraits de *The Planets* (« *Les Planètes* », 1918) de Gustav Holst (1874-1934), dont on retrouve l'empreinte dans la bande originale composée par John Williams. Le monteur de la musique peut aussi être amené à monter des morceaux préexistants, comme des chansons, destinés quant à eux à demeurer dans la version finale du film : ces morceaux ne seront donc pas remplacés par le compositeur, et cohabiteront avec la bande originale du film ; on parle alors de « musique additionnelle »³¹.

30 « [...] temp tracks are a quite recent development. In the early days of Cinema, for instance, the Golden Age in Hollywood, there would be no temp track since it is a practice, which was technologically unavailable. » — Paula Flach, *Film Scoring Today: Theory, Practice and Analysis* [en ligne], Mémoire de Maîtrise (sous la dir. de Peter Larsen), Université de Bergen, 2012, pp. 40-41.

31 Dans un film, les musiques additionnelles sont des morceaux qui n'ont pas été composés spécifiquement pour le film, comme une chanson préexistante. On entend aussi parfois par « musique additionnelle » un morceau qui n'a pas été composé par le compositeur de la musique originale, mais par un de ses assistants ou co-compositeurs.

Comme l'explique Richard Davis, « chaque compositeur a une approche légèrement différente du processus de composition d'une musique de film. Mais il y a deux styles distincts, que nous appellerons « l'approche traditionnelle » et « l'approche non-traditionnelle. » »³². Ces deux approches se distinguent par leurs méthodes de travail, et nous pouvons envisager l'approche « non-traditionnelle » comme celle de la « nouvelle génération » de compositeurs, qui depuis la fin des années 1990 s'oriente et évolue dans un processus tout-numérique. C'est cette seconde approche qui va nous intéresser au premier chef, mais pour bien la comprendre, il faut dans un premier temps nous attarder sur l'approche « traditionnelle », qui est celle de compositeurs comme Bernard Hermann, John Williams, ou encore Ennio Morricone.

Dans ce cadre « traditionnel », donc, le monteur de la musique se présente à la *Spotting Session*, et rédige des *Spotting Notes*, ou « notes de repérage », afin de préparer ensuite des *Timing Notes*, c'est-à-dire des « notes de minutage ». Les *Spotting Notes* renseignent sur le début et la fin de chaque *cue*, et récapitulent toutes les instructions et les informations données par le réalisateur et le compositeur, comme par exemple l'instrumentation, l'entrée d'un thème à un moment précis, etc. Le monteur de la musique crée ensuite une « *Master Cue List* » ou « *Music Summary* », soit un « sommaire musical », c'est-à-dire une liste de tous les *cues* qui doivent être composés, et leur position dans le film. Ce sommaire permet tout au long du processus d'assurer le suivi de toutes les étapes qui doivent être franchies, et de s'assurer de l'emplacement et de la disponibilité de tous les éléments nécessaires au passage d'une étape à la suivante. Une note de minutage est ensuite rédigée pour chaque *cue* : il s'agit de détailler tous les plans, toutes les coupes, et toutes les lignes de dialogue prononcées dans l'intervalle de temps couvert par chaque *cue*. Ces *Timing Notes* permettent notamment de choisir des points de

32 « Every composer has a slightly different approach to the process of writing a film score. But there are two distinct styles that [...] I will call the *traditional approach* and the *non-traditional approach*. » — Richard Davis, *op. cit.*, p. 95.

synchronisation pour la musique, et de trouver le tempo adéquat pour chaque séquence.

La méthode « non-traditionnelle », quant à elle, consiste à jouer en même temps que le film, et à enregistrer des maquettes pour chaque *cue*. Cette pratique a commencé à se développer avec les débuts de la MAO, et s'est implantée en même temps que la numérisation de l'industrie et de la distribution cinématographiques. Ainsi, les *Timing Notes* qui renseignent sur tous les plans, leur enchaînement et le dialogue contenus dans une séquence ne sont plus utiles, puisque le compositeur les voit et les entend en temps réel. Il peut donc se contenter d'une liste de points de synchronisation qu'il souhaite mettre en place, tout en gardant à l'esprit les décisions qui auront été prises avec la réalisation et la production lors de la session de repérage.

3. La composition, l'orchestration, et la préparation de la session d'enregistrement de la bande originale

Le compositeur travaille d'abord sur une ébauche musicale, c'est-à-dire une version condensée de chaque *cue*. Auparavant, une ébauche était écrite sur des feuillets de papier à musique de quelques portées. Nous pouvons dire qu'une ébauche est comme la « carte d'un *cue* »³³ plus ou moins détaillée, c'est-à-dire plus ou moins orchestrée. Aujourd'hui, une ébauche prend la forme d'un fichier MIDI généré d'après la maquette enregistrée par le compositeur sur sa STAN. Ces fichiers MIDI sont transmis à un assistant qui doit quantifier chaque piste, tout en s'assurant que la matière qu'il transmettra ensuite à l'orchestration soit bien conforme à celle jouée et enregistrée par le compositeur.

Tristan Noon³⁴ décrit avec précision le passage de la musique écrite par le compositeur à la session d'enregistrement. Dans un premier temps, le

33 « A sketch is like a map of the cue. » — Richard Davis, *op. cit.*, p. 122.

34 Tristan Noon, *op. cit.*, pp. 10-19.

compositeur produit une maquette MIDI plus ou moins aboutie dans sa STAN grâce à des instruments virtuels³⁵. Ces maquettes permettent d'avoir une idée relativement précise du résultat final de chaque *cue*, une fois qu'il aura été orchestré, enregistré et mixé. Une maquette MIDI permet donc à la réalisation et à la production de valider ou non chaque *cue*. Une fois qu'un *cue* est validé, il est transmis à un assistant, qui va devoir d'une part réaliser un export de chaque *cue* sous la forme d'un fichier MIDI quantifié, c'est-à-dire parfaitement aligné sur la grille musicale, et d'autre part un export audio de chaque *stem* pour l'ingénieur du son en charge de la séance d'enregistrement de la musique.

Un *stem* est un export audio qui peut être considéré comme un sous-groupe réunissant plusieurs pistes sous la forme d'une sommation. Les différentes manières d'exporter un *cue* sous la forme de *stems* peuvent être réduites à trois possibilités :

(1) Exporter chaque piste séparément avec un même point d'entrée et un même point de sortie, ce qui induit un nombre conséquent de pistes et de fichiers à gérer, mais qui permet une grande flexibilité durant la séance d'enregistrement et éventuellement durant le mixage.

(2) Exporter les pistes par groupes ou familles d'instruments, comme par exemple tous les bois ensemble, tous les cuivres ensemble, etc., ce qui diminue le nombre de pistes et de fichiers audio, mais réduit aussi notre marge de manœuvre.

(3) Exporter les pistes sous la forme de sous-groupes « utiles », afin d'avoir assez de flexibilité durant l'enregistrement et éventuellement au mixage (dans le cas où on utilise aussi ces *stems*) sans pour autant exporter toutes les pistes individuellement. Cette troisième méthode est à laisser à la discrétion du compositeur et de l'ingénieur du son en charge de la session d'enregistrement, en fonction des besoins de

³⁵ Les notions d'« instrument virtuel » et de « maquette MIDI » sont abordées dans le détail un peu plus tard.

chacun. On pourra par exemple séparer les articulations longues des articulations courtes, et les solistes des *tutti* pour chaque famille d'instrument. Il est aussi possible de séparer chaque famille d'instrument par hauteur, ce qui donnerait par exemple pour les cordes trois *stems*, avec les « cordes aiguës », les « cordes médiums », et les « cordes graves ». Nous pourrions aussi exporter six *stems* en combinant cette logique avec la précédente, soient les exports suivants : « cordes aiguës longues », « cordes aiguës courtes », « cordes médiums longues », « cordes médiums courtes », « cordes graves longues » et « cordes graves courtes ».

Toutefois, Tristan Noon précise qu'il est préférable de livrer toutes les pistes séparément pour la session d'enregistrement, ce qui permet de ne pas avoir à refaire d'exports pour le mixage de la musique³⁶. Peut-être que cette approche est à nuancer et à affiner : nous pouvons par exemple exporter les éléments qui seront livrés tels quels pour le mixage (percussions, synthétiseurs, guitares, etc.) sous la forme de pistes séparées, et les éléments qui seront enregistrés, et demeureront à l'état de maquettes, comme les instruments de l'orchestre, sous la forme de sous-groupes plus ou moins détaillés.

Une fois que les *stems* lui ont été livrés, l'ingénieur du son peut préparer les sessions nécessaires à l'enregistrement de chaque *cue*. Dans la majorité des cas, c'est le logiciel *ProTools* qui est utilisé. Il est préférable de créer un fichier de session par *cue*, mais certains préfèrent créer une session par bobine, ou encore une session pour l'intégralité du film. Quoiqu'il en soit, tous les *stems* sont importés, ainsi que le fichier MIDI, qui permet d'ajouter la carte de tempo, ou « *tempo map* », issue de la STAN du compositeur. Cette dernière transporte les valeurs et les variations de *tempo* et de signature rythmique, afin que la grille de la session *ProTools* et le transport soient alignés et synchronisés avec la partition du *cue*. En d'autres termes, la *tempo map*

36 Tristan Noon, *op. cit.*, p. 14.

transporte tous les paramètres qui permettent d'assurer la synchronisation de la session avec la partition et l'image. Elle sera également utile pour générer le clic du métronome qui permettra au chef d'orchestre et aux musiciens de jouer ensemble, synchrones à l'image, c'est-à-dire de respecter les points de synchronisation et les *tempi* prévus par le compositeur. Les pistes de clic sont en général fabriquées par le monteur de la musique du film, à qui revient plus globalement la responsabilité de la synchronisation de la musique à l'image à toutes les étapes de fabrication du film. Il prépare donc les pistes de tempo, mais aussi les repères visuels qui permettront au chef d'orchestre d'anticiper les points de synchronisation entre la musique et l'image.

Nous avons vu que chaque *cue* génère des *stems* audio, mais aussi un fichier MIDI. Ce dernier va être utile pour l'étape d'orchestration, qui d'après Tristan Noon, a changé de manière significative durant ces dernières années³⁷. Elle constitue sans doute l'étape qui a subi le plus de mutations dans le processus de fabrication d'une musique de film, qui s'est métamorphosé dans son ensemble depuis que l'informatique musicale s'est substituée au papier. Là où l'orchestration était réalisée d'après une ébauche écrite sur six ou huit portées, elle consiste aujourd'hui à importer un fichier MIDI extrait de la STAN du compositeur, et à l'orchestrer grâce à un logiciel de notation musicale, comme *Finale* ou *Sibelius*. Ainsi, la tâche de l'orchestrateur de la « nouvelle génération » consiste le plus souvent à « convertir » un fichier MIDI généré par la STAN du compositeur en partition³⁸. Cette conversion sera plus ou moins transparente et plus ou moins longues suivant les cas : l'orchestrateur devra compléter et affiner quelques éléments et faire plus ou moins de propositions musicales et instrumentales en fonction du degré de détails de la maquette originale proposée par le compositeur.

37 « The term « orchestration » is (in my opinion) one that has changed quite significantly in recent years. » — Tristan Noon, *op. cit.*, p. 8.

38 « I refer to the new-style orchestrator as one which takes MIDI files from the composer's sequencer of choice and converts it onto score form. » — Tristan Noon, *op. cit.*, p. 8.

Une fois que le conducteur pour un *cue* est terminé et validé par le compositeur, il est envoyé au(x) copiste(s). L'étape de la copie consiste à finaliser le conducteur, mais surtout à éditer les parties séparées destinées à tous les musiciens qui enregistreront chaque *cue*. Dès qu'une partie séparée est copiée, elle est vérifiée par un correcteur, qui la confronte au conducteur avant qu'elle ne soit imprimée. Pour les films à gros budget, un « bureau », le « *Music Preparation Office* », s'occupe en général de la copie, mais surtout de la préparation de la séance d'enregistrement de la musique : il faut vérifier que tous les *cues* soient orchestrés, copiés et imprimés dans le bon ordre, afin que tout le « matériel » soit prêt pour chaque jour d'enregistrement. Lorsque tout est prêt pour l'enregistrement de la musique, un documentaliste s'assure que chaque musicien a bien toutes les partitions dont il a besoin.

Le travail de l'orchestrateur a beaucoup évolué avec la numérisation du processus de fabrication d'une musique de film, mais c'est aussi le cas de celui de l'*assistant*. Jusqu'aux années 1990, l'assistant était avant tout chargé de s'occuper de tâches administratives, mais à l'avènement de l'informatique musicale, conjointe à la numérisation de l'industrie cinématographique, les tâches de l'assistant se sont orientées vers la gestion des outils technologiques utilisés par le compositeur, et par ricochets se sont étendues à des tâches plus musicales et créatives, notamment de co-composition, d'orchestration, de copie et de montage.

L'arrivée du numérique a entraîné une accélération de la post-production cinématographique, et les temps de travail ont été réduits : le rôle principal de l'assistant, et plus largement de toute l'équipe qui entoure le compositeur, est de lui faire gagner du temps, c'est-à-dire de le décharger autant que faire se peut de tâches qui ne concernent pas la composition en elle-même. Parmi ces nombreuses tâches, il y a bien entendu le recrutement des musiciens et l'organisation pratique de la séance d'enregistrement, mais nous ne les détaillerons pas ici. Nous sommes à présent prêts à enregistrer la musique du

film : tous les *cues* ont été composés, orchestrés, copiés et imprimés, les sessions *ProTools* ont été préparées, et les musiciens ont été engagés.

4. La séance d'enregistrement et le mixage de la musique

L'enregistrement de la musique composée pour un film et son mixage se situent en aval de notre problématique. Nous nous contenterons donc de parcourir rapidement ces deux dernières étapes, qui ne doivent évidemment pas être négligées. En fin de compte, les différentes étapes de la réalisation d'une musique de film sont caractérisées par les mêmes trois grandes phases de fabrication d'un film : la pré-production consisterait à la composition musicale, à l'orchestration, à la copie et à la préparation de la séance d'enregistrement ; la production à la séance d'enregistrement elle-même ; et enfin la post-production au mixage.

Durant la séance d'enregistrement, le monteur de la musique note pour chaque prise toutes les informations qui lui semblent utiles pour le montage final de chaque *cue*. Parfois, il est nécessaire de déplacer, d'éditer, voire même de reconstruire complètement un *cue* avec des éléments d'autres *cues* durant le mixage, suite à des changements décidés après la séance d'enregistrement.

Une fois que la musique est enregistrée, elle est mixée avant d'être envoyée à l'auditorium, où la bande son du film est mixée dans son intégralité. Pour le cinéma, la musique est livrée sous la forme d'un mixage multi-canal et de *stems*. Les *stems* sont à niveau, ce qui signifie que leur sommation est égale au mixage multi-canal définitif de la musique. Ils offrent cependant plus de latitude au mixeur pour ajuster tel ou tel élément musical par rapport au reste de la bande sonore du film ; ils ne sont donc pas systématiquement utilisés, mais leur livraison est obligatoire.

Le mixage de la bande son du film se déroule en général en deux temps : avant de commencer à mixer tous les éléments sonores, le mixeur

réalise une première passe de « mixage des paroles », ou de « *pre-dubbing* », qui consiste à travailler uniquement sur le timbre, la clarté, la propreté et la spatialisation des dialogues. La musique est mixée en parallèle de cette première étape. Une fois que les dialogues, la musique, le montage son et les bruitages sont prêts, ils sont mixés ensemble. Il n'est pas rare qu'à cette ultime étape de la post-production, la musique soit déplacée ou éditée. Comme nous l'avons vu, c'est en général le monteur de la musique du film qui est responsable de ces ajustements, et le compositeur n'est [presque] plus concerné par le mixage : il a accompli la tâche qui lui avait été confiée. Bien évidemment, ce dernier assiste épisodiquement au mixage, afin de donner son avis, bien qu'une fois encore, cela dépende de la collaboration et de la relation qu'il entretient ou non avec la réalisation et/ou la production.

Une fois que le mixage du film est achevé — autrement dit, une fois que le film est terminé —, le monteur de la musique doit rédiger une « Feuille de montage », qui récapitule toute la musique qui a finalement été utilisée dans le film, qu'elle soit originale ou additionnelle. Cette feuille de montage, ou « *Music Clearance Sheet* », ou encore « *Cue Sheet* »³⁹, est utilisée pour la gestion des droits d'auteurs des ayants-droits des différentes musiques qui constituent l'ensemble de la bande musicale du film. Une ultime étape, que nous ne détaillerons pas ici, consiste à éventuellement réaliser un album de la bande originale du film. Cela implique en général une étape de montage, puis un mixage et un *mastering* stéréophoniques des pistes qui auront été retenues pour y figurer.

Pour conclure, nous avons décrit et détaillé une chaîne de production « hollywoodienne », c'est-à-dire réservée à des films à gros budget, et disposant de moyens humains et financiers conséquents. Il s'agit finalement d'une chaîne « idéale » vers laquelle la majorité des productions tendent, sans

³⁹ Attention, comme nous avons déjà eu l'occasion de l'évoquer, le terme de « *cue sheet* » est utilisé pour désigner plusieurs documents : il faut donc être vigilant au contexte dans lequel il est employé. Par exemple, certains compositeurs parlent de « *cue sheet* » pour parler de leurs « *Timing Notes* ».

forcément avoir les moyens d'y aspirer. Dans la majorité des cas, le compositeur doit endosser plusieurs rôles, tout comme ses collaborateurs, s'il a la chance de ne pas travailler seul. Par exemple, un compositeur peut se charger lui-même de l'orchestration et déléguer uniquement la copie des partitions, ou son assistant peut être aussi copiste et monteur de la musique du film, etc. Chaque film est un prototype, et dans cette mesure, toutes les configurations et toutes les méthodes de travail peuvent être envisagées.

II. Le studio de composition : le MIDI et les instruments virtuels

A. Le MIDI : rappels et pré-requis

Le *Musical Instrument Digital Interface* (« Interface Numérique des Instruments de Musique »), abrégé « MIDI », est à la fois un protocole de communication et un format de fichier dédié à la musique. En tant que protocole, il est notamment utilisé pour établir la communication entre des instruments dits « électroniques », comme des synthétiseurs, mais plus largement entre toutes sortes de contrôleurs et de logiciels de MAO. En tant que structure de fichier, le MIDI s'est imposé au fil des années et de ses évolutions comme un standard pour l'échange de données musicales entre plusieurs utilisateurs et plusieurs plateformes. Il est aujourd'hui omniprésent dans la MAO, que ce soit au niveau des logiciels de composition, de notation musicale, de traitement de l'audio, ou encore des contrôleurs et des fichiers qui sont échangés par de nombreux créateurs et producteurs de musique.

Après un bref historique de la norme MIDI, nous nous intéresserons en profondeur à la structure d'un fichier MIDI, puisque c'est sous cette forme que les données musicales transitent entre une STAN et un logiciel de notation musicale. Enfin, nous aborderons l'utilisation du protocole MIDI dans une STAN.

1. Petite histoire du MIDI

Les caractéristiques de la norme MIDI ont été définies pour la première fois en 1982 dans une publication de la *MIDI Manufacturers Association* (MMA, ou « Association des Fabricants de matériel MIDI »). Avant 1995, cette spécification est déclinée en cinq documents distincts : le *Detailed MIDI*

Specification Document Version 4.2 (« Spécification MIDI détaillée, version 4.2 »), incluant le *MIDI timecode* (TC)⁴⁰, le *MIDI Show Control* (« MSC »), le *MIDI Machine Control* (« MMC »), le *Standard MIDI Files* (« SMF »), et enfin le *General MIDI Level 1* (« GM »). En 1995, ces cinq documents sont compilés en un seul, *The Complete MIDI 1.0 Detailed Specification* (« La Spécification MIDI complète et détaillée 1.0 »), qui a depuis été régulièrement mis à jour. Comme nous l'avons déjà évoqué, nous n'allons pas nous intéresser à l'ensemble de la norme MIDI, mais plus spécifiquement à la structure d'un fichier MIDI standard (SMF). Nous survolerons tout de même les spécifications de la norme MIDI en temps que protocole de communication : pour le dire autrement, notre étude du SMF doit être appréhendée du côté du « développeur », et celle du MIDI en tant que protocole de communication du côté de l'« utilisateur ».

Il nous paraît important de préciser dès à présent que tous les documents officiels relatifs aux caractéristiques de la norme MIDI ont été rédigés en langue anglaise. Nous avons pensé dans un premier temps traduire systématiquement tous les termes propres au MIDI, mais il nous a semblé finalement plus judicieux de les conserver dans leur langue originale, afin que nos travaux puissent être facilement confrontés à ces documents et plus largement à l'ensemble de la littérature dédiée au MIDI.

2. La structure d'un fichier MIDI (SMF)

Le SMF permet de transférer des données MIDI entre plusieurs utilisateurs, plusieurs machines, ou encore plusieurs programmes informatiques, puisqu'il s'agit d'un format extrêmement ouvert et commun aux principaux systèmes d'exploitation du marché que sont *macOS*, *Windows* et

⁴⁰ Le *timecode* (TC) s'exprime en heures, minutes, secondes et images : il permet d'assurer la synchronisation entre plusieurs machines en donnant une adresse unique à chaque image d'un programme audiovisuel.

Linux. Contrairement à la plupart des autres types de fichiers musicaux, un fichier MIDI n'encode pas de données audio, comme c'est par exemple le cas du format WAVE : un fichier MIDI est donc à la fois dense et extrêmement léger, ce qui le rend particulièrement approprié pour le stockage et l'exploitation sur disque dur, mais moins adéquat pour un chargement en mémoire pour un accès rapide. C'est pour cela qu'un SMF n'est pas utilisé tel quel dans une STAN, et qu'il est apparenté à un flux MIDI, c'est-à-dire qu'il est exploité comme un protocole de communication une fois qu'il a été importé.

Un fichier MIDI est structuré par une série de blocs de données, ou « *chunks* », qui se déclinent à ce jour en deux types : le « bloc d'en-tête » ou « *Header Chunk* », qui contient les propriétés globales du fichier MIDI, et le(s) « bloc(s) de piste », ou « *Track Chunk(s)* », qui contiennent quant à eux des événements MIDI qui décrivent l'ensemble du contenu musical du SMF. Ainsi, un fichier MIDI valide débute par un bloc d'en-tête, suivi par un ou plusieurs bloc(s) de piste.

En termes de syntaxe, chaque bloc débute par quatre caractères ASCII⁴¹, qui sont immédiatement suivis par un multiplét⁴², ou « *byte* », de trente-deux bits, qui code le nombre d'octets de données qui sont contenus dans ce bloc ; il s'agit de la donnée de longueur *<length>*. Notons tout de suite que dans le cas

41 « ASCII » est l'abréviation de « *American Standard Code for Information Interchange* », qui signifie « Code Américain Normalisé pour l'Échange d'Information ». Créé dans les années 1960, l'ASCII est la norme informatique de codage de caractères la plus répandue et la plus influente à ce jour.

42 Un multiplét est la plus petite unité « logiquement » adressable par un programme sur un ordinateur. Aujourd'hui, les besoins d'une structure commune pour le partage de l'information ont conduit à une généralisation des multipléts de 8 bits. Nous avons donc tendance à confondre la notion de multiplét avec celle d'octet. Dans les faits, un *octet* est un multiplét contenant exactement 8 bits. En anglais, un multiplét est traduit par « *byte* », tandis qu'un octet est défini comme un « *eight-bit byte* », ce qui prête moins à confusion. Le *bit*, ou *élément binaire*, est quant à lui l'unité la plus simple dans un système de numération, et ne peut prendre que deux valeurs, désignées le plus souvent par les chiffres 0 et 1.

d'un SMF, et plus largement dans celui de la norme MIDI, les valeurs qui sont codées sur plusieurs octets sont écrites au format gros-boutiste⁴³, c'est-à-dire que leur MSB⁴⁴ vient en premier. Chaque bloc débute donc par huit octets qui nous renseignent sur son type, et nous donnent la taille des données qui lui sont associées. Les octets de 0 à 3 inclus nous renseignent sur le type du bloc (on parle d'« *identifieur* »), tandis que les octets de 4 à 7 indiquent le nombre d'octets nécessaires au codage des données associées au bloc, autrement dit la longueur de ces données (on parle cette fois-ci de « *chunklen* »). Les huit octets qui définissent le *<type>* du bloc et la longueur des données du bloc *<length>* ne sont pas inclus dans le calcul de la longueur du bloc : ainsi, un bloc dont la longueur serait de neuf octets occuperait en réalité $8 + 9 = 17$ octets sur notre disque dur de stockage.

Octet n°	0	1	2	3	4	5	6	7	8	9	[...]	(fin)	
	<i><type></i>				<i><length></i>				<i><data></i>				
	<i><type></i>				<i><longueur></i>				<i><données></i>				

Figure 2: Représentation de la syntaxe d'un bloc

43 En informatique, certaines données, comme les nombres entiers, peuvent être représentées sur plusieurs octets. L'ordre dans lequel ces octets sont organisés en mémoire ou dans une communication est désigné par les termes « boutisme » ou « endianisme » (en anglais, on parle de « *endianness* »). De manière analogue à certains langages humains qui s'écrivent de gauche à droite ou de droite à gauche, il existe une alternative majeure à l'organisation des octets représentant une donnée : l'orientation « gros-boutiste » ou « grand-boutien », qui place le MSB en premier ou en tête, c'est-à-dire à gauche de l'octet, et l'orientation « petit-boutiste » ou « petit-boutien », qui place cette fois-ci le LSB en tête ; autrement dit, le MSB se trouve alors à droite de l'octet.

44 Le MSB désigne le *bit de poids fort*, ou « *Most Significant Bit* », qui dans une représentation binaire donnée, a la plus grande valeur. *A contrario*, le LSB, ou « *Least Significant Bit* », est pour un nombre binaire dans une représentation donnée la moindre valeur, c'est-à-dire la moins significative.

Cette structure de bloc rudimentaire offre d'une part la possibilité de créer facilement de nouveaux types de blocs, et d'autre part d'ignorer des blocs identifiés en tant que blocs, mais dont le type ne serait pas (re)connu par notre programme. Pour le dire autrement, la norme MIDI a dès sa création été tournée vers l'avenir et un développement ouvert, tout en conservant le souci d'une rétro-compatibilité sur le long terme. Le cas le plus évident est celui d'un type de bloc créé alors qu'un programme n'est plus mis à jour, et n'est donc pas capable de l'interpréter : notre programme sera toujours en mesure d'ouvrir le fichier MIDI qui contient ce nouveau type de bloc, de lire les données contenues dans les blocs dont il connaît le type, tout en ignorant le bloc qui lui est inconnu ; il sera néanmoins capable de comprendre qu'il s'agit d'un bloc, même s'il ne peut le déchiffrer et accéder aux données qu'il contient.

2.1. Le « *Header Chunk* » ou le bloc d'en-tête

Comme nous l'avons déjà évoqué, le bloc d'en-tête, d'une longueur fixe de quatorze octets, renseigne sur la structure du fichier MIDI, et contient les informations globales du SMF. Un fichier MIDI standard doit nécessairement débiter par un bloc d'en-tête, et il ne doit logiquement n'en présenter qu'une seule et unique occurrence. La syntaxe d'un bloc d'en-tête est la suivante :

```
<Header Chunk> = <chunk type><length><format><ntrks><division>  
<Bloc d'en-tête> = <type de bloc><longueur><format><nombre de pistes><division>
```

Figure 3: Syntaxe d'un bloc d'en-tête

Le type d'un bloc d'en-tête est donné par les quatre caractères ASCII « MThd », tandis que la section qui comprend ses données est constituée de trois mots de seize bits : le *<format>*, le nombre de bloc(s) de piste *<ntrks>* contenus dans le fichier, et la *<division>* [temporelle] qui régit les évènements encapsulés dans le fichier MIDI.

2.1.1. Les formats d'un fichier MIDI standard

Le *format* d'un SMF nous renseigne sur son organisation, mais aussi sur le rapport qui existe ou non entre les blocs de piste qui le composent. La variable <format> peut prendre trois valeurs (x0000, x0001, x0002), puisqu'il existe trois formats de fichier MIDI distincts :

a. Le *Format 0*

Le SMF contient une seule piste qui peut être multi-canal, c'est-à-dire un seul bloc de données où se mélangent les évènements de plusieurs canaux MIDI. Autrement dit, toutes les données sont combinées en un seul bloc de piste.

b. Le *Format 1*

Le fichier MIDI contient au moins deux blocs de piste, dont le nombre est renseigné par la variable <ntrks> : la première piste est réservée à des évènements globaux, comme des évènements de *tempo* ou de signature rythmique⁴⁵, utiles à la synchronisation des pistes entre elles, tandis que les suivantes contiennent des données de notes et divers contrôles liés à leur interprétation. Les pistes d'un fichier MIDI de *Format 1* sont jouées simultanément et synchrones, et sont réparties par canal et/ou par sortie MIDI : nous pouvons ici faire l'analogie avec le fonctionnement d'un enregistreur multipistes. C'est ce format qui est aujourd'hui le plus utilisé, puisqu'il permet d'enregistrer chaque partie instrumentale sur une piste séparée, ce qui facilite toutes les opérations d'édition que nous voudrions entreprendre.

45 Nous rappelons que l'*Annexe 1* propose un lexique et quelques définitions sommaires des termes musicaux les plus utiles et les plus répandus, comme *tempo*, *signature rythmique*, mais aussi la dénomination des différentes valeurs rythmiques, des modes de jeu, etc.

c. Le *Format 2*

Le fichier MIDI contient une ou plusieurs pistes MIDI successives et indépendantes, ce qui signifie que le fichier contient plusieurs plages de données, mais que contrairement au *Format 1*, elles ne sont pas destinées à être jouées simultanément, mais l'une après l'autre. De la même manière que nous avons pu établir une analogie entre le *Format 1* et le fonctionnement d'un enregistreur multipistes, nous pouvons comparer le *Format 2* à une boîte à rythmes, dans laquelle chaque piste serait un échantillon. Notons tout de même que ce dernier format est peu utilisé.

2.1.2. Le nombre de pistes <ntrks> d'un fichier MIDI

Le nombre de blocs de piste est donné par la variable <ntrks>. Pour un fichier MIDI au *Format 0*, la valeur <ntrks> est toujours égale à 1 ; elle est variable pour les fichiers de *Format 1* et de *Format 2*. La spécification MIDI ne prévoit pas de limitation du nombre de pistes que peut contenir un SMF, mais ce n'est en général pas le cas des STAN : si un fichier MIDI, dont le nombre de blocs de piste dépasse le nombre de pistes autorisées, est importé dans une STAN, alors cette dernière se contente d'ignorer les pistes qui dépassent cette limite, et importe les autres.

2.1.3. La <division> temporelle d'un fichier MIDI

La dernière variable du bloc d'en-tête, <division>, spécifie la nature et la résolution de l'intervalle de temps Δt , qui comme nous le verrons, définit l'écart de temps entre deux événements MIDI consécutifs. La nature de cet intervalle peut découler de la métrique musicale ou du *timecode* (TC), qu'il soit

MIDI ou SMPTE⁴⁶. Si la division du SMF est liée à la métrique musicale, alors les écarts de temps entre les évènements dépendent du tempo, tandis que si la division temporelle est liée au TC, alors les Δt sont absolus, et donc indépendants du *tempo* et de ses variations.

Si le MSB de <division> est égal à 0, alors la base de temps du fichier MIDI est la métrique musicale. Les bits 14 à 8 nous renseignent alors sur le nombre de subdivisions d'une noire, ou de battements par noire : on parle de PPQN (« *Pulses Per Quarter Note* »). Dans ce cas, les bits de 7 à 0 ne sont pas utiles. Par exemple, si la division du fichier MIDI est égale à 960 PPQN, qui est la valeur la plus commune puisqu'elle est divisible par deux et par trois, et permet donc de représenter facilement des triolets, des notes pointées, etc., alors l'écart de temps entre deux croches dans ce fichier sera de 480 PPQN.

Bit n°	15	14 à 8	7 à 0
0		Résolution musicale (PPQN)	(inutilisés)
1		Cadence-image négative	Subdivision de l'image

Figure 4: Structure de l'octet <division> d'un SMF

46 La SMPTE (*Society of Motion Picture and Television Engineers*, « Société des Ingénieurs du Cinéma et de la Télévision ») développe des standards vidéo qui sont notamment utilisés par la télévision et le cinéma. Ces standards incluent entre autres tous les formats de transmission, des interfaces physiques pour la transmission de signaux de télévision et les données associées, comme le *time code* SMPTE, ou encore le *MXF*.

A contrario, si le MSB de <division> est égal à 1, alors la base de temps du SMF est le TC. Dans ce cas, les bits 14 à 8 codent quatre valeurs possibles sous la forme d'entiers négatifs, qui correspondent à l'une des quatre cadences d'image définies dans les standards SMPTE et MIDI : -24, -25, -29 (qui correspond à 30 *drops* par image⁴⁷) et -30. Le second octet indique la résolution temporelle par image, c'est-à-dire la subdivision de l'image. Une image peut-être subdivisée par 4, qui correspond à la résolution du *MIDI Timecode*, 8, 10, 80, soit la résolution du *bit SMPTE*, ou encore par 100. Ainsi, la norme MIDI prévoit une spécification des pistes basée sur le TC, mais aussi indirectement une résolution basée sur la milliseconde, qui peut être obtenue en fixant la cadence-image à 25 images par seconde, et une subdivision de l'image par 40 ; dans ce cas, on a bien l'écart de temps minimum $\Delta t_{min} = 1$ ms.

En MIDI, la plus petite division du temps, qu'elle soit exprimée en un PPQN ou en subdivisions de l'image, est appelée le « *tick* ». Pour le dire autrement, si notre base de temps est la métrique musicale, alors 1 *tick* correspond à 1 PPQN, tandis que si notre base de temps est le TC, ou par extension la milliseconde, alors 1 *tick* correspond à une subdivision de l'image.

47 Le format vidéo NTSC (*National Television System Committee*, « Comité du Système de Télévision Nationale ») est un standard de codage analogique de la vidéo en Amérique du Nord. Il présente la particularité d'avoir une cadence-image de 30 images par seconde en noir et blanc et de 29,97 images par seconde en couleurs. Il y a donc deux types de TC comptant à 30 images par seconde pour le format NTSC : le format « *non-drop frame* » et le format « *drop frame* ». Le format « *non-drop frame* » est très simple : à chaque image est associé un TC, autrement dit, le TC s'incrémente à chaque image sans compensation ; c'est le cas le plus fréquent. Le format « *drop frame* », qui n'est logiquement utilisé qu'en présence d'une vidéo au format NTSC, compense le fait que la cadence-image du NTSC est de 29,97 images par seconde, soit 0,3 image par seconde plus lent que le nombre entier le plus proche pour une cadence de 30 images par seconde. Seulement, le TC ne peut être représenté que par des nombres entiers, donc certains nombres du TC sont périodiquement sautés. De cette manière, le TC correspond toujours à la seconde près à la vidéo qui est lue. Dans le cas de la spécification du MIDI, l'entier négatif « -29 » correspond donc à un TC au format « *drop frame* », qui pourra être utilisé dans le cas où un fichier vidéo au format NTSC-couleur doit être synchronisé avec notre SMF.

2.2. Le(s) « *Track Chunk(s)* » ou le(s) bloc(s) de piste

Les « *Track Chunks* », ou « blocs de piste », contiennent tous les évènements qui permettent de décrire le contenu musical du SMF. Ces derniers se déclinent en trois types : (i) les *MIDI Control Events* (« évènements de contrôle MIDI »), (ii) les *System Exclusive Events*, abrégés *SysEx Events* (« évènements système exclusif »), et enfin (iii) les *Meta Events* (« évènements méta »). À chacun de ces évènements est associé un intervalle de temps, un « *delta time* », ou « Δt », qui correspond à la quantité de temps écoulée depuis l'évènement précédent ; ce Δt est donné en *ticks*. La syntaxe d'un bloc de piste est identique, quel que soit le format du fichier MIDI (0, 1 ou 2) dans lequel il est encapsulé :

```
<Track Chunk> = <chunk type><length><MTrk event(s)>
```

```
<Bloc de Piste> = <type de bloc><longueur><évènement(s) de piste>
```

Figure 5: Syntaxe d'un bloc de piste

Comme pour le bloc d'en-tête, le <type> est donné par quatre caractères ASCII, qui sont pour le bloc de piste « MTrk ». <MTrk event(s)> correspond aux évènements contenus dans le bloc ; une piste doit contenir au moins un évènement, mais elle peut en avoir un [très] grand nombre. Nous rappelons qu'au *Format 1*, le premier bloc de piste est une piste « globale », et qu'elle ne contient que les évènements méta qui s'appliquent à l'ensemble du SMF, comme des variations de *tempo*, de signature rythmique, ou encore d'armure, mais pas de données de notes à proprement parler. En effet, comme toutes les pistes d'un SMF de *Format 1* sont jouées simultanément, elles doivent suivre une « carte de tempo », ou « *tempo map* », décrite dans la première piste grâce à l'évènement méta de type 81. Quant aux données de notes, qui s'apparentent à une performance musicale, elle sont réparties entre la seconde piste et les suivantes. Enfin, bien qu'un bloc de piste puisse

accueillir trois types d'évènements différents, ces derniers partagent une syntaxe commune, donnée ci-dessous :

```
<MTrk event> = <delta time><event>
```

```
<Évènement de piste> = < $\Delta t$ ><évènement>
```

Figure 6: Syntaxe d'un évènement MIDI

2.2.1. Le <delta time> ou Δt

La variable <delta-time> est une quantité de longueur variable⁴⁸ de un à quatre octets. Elle représente la « quantité » de temps écoulée depuis le déclenchement de l'évènement précédent. Pour le dire autrement, il s'agit de l'intervalle de temps entre deux évènements donné en *ticks*, dont la résolution est fonction de la base temporelle renseignée par la variable <division> du bloc d'en-tête. La plupart du temps, les évènements musicaux sont assez proches les uns des autres, ce qui permet de coder le Δt qui les sépare sur un seul octet. Notons que si un évènement a lieu au début d'une piste, il sera précédé d'un <delta-time> égal à zéro ; il en est de même pour plusieurs évènements qui doivent être déclenchés simultanément.

2.2.2. La quantité de longueur variable (VLQ)

La « quantité de longueur variable », ou VLQ pour « *Variable Length Quantity* », est une représentation en base-128 de nombres entiers non-signés introduite et définie en même temps que la norme MIDI et le SMF⁴⁹ afin d'optimiser l'usage des espaces de stockage. Autrement dit, la VLQ est un système de codage universel qui utilise un nombre arbitraire d'octets, c'est-à-dire le nombre strictement nécessaire, pour représenter un entier plus ou

⁴⁸ Le principe de la quantité de longueur variable (VLQ) est expliqué ci-après (2.2.2.).

⁴⁹ La spécification du SMF donne quelques exemples de nombres entiers codés en VLQ, qui sont donnés en *Annexe 2*.

moins grand. En MIDI, la longueur d'un nombre codé en VLQ est comprise entre un et quatre octet(s). La structure générale d'un octet VLQ est simple : le MSB permet de déterminer si l'octet VLQ est le dernier octet codant l'entier, ou si un ou plusieurs autres octets sont encore nécessaires, et suivent. Puis, les sept autres bits codent un nombre. Pour finir, on additionne les nombres codés par les différents octets VLQ, afin d'obtenir l'entier non-signé. La figure ci-dessous récapitule la structure d'un octet VLQ :

7	6	5	4	3	2	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
A	B_n						

Figure 7: Structure d'un octet VLQ

Si $A = 0$, alors l'octet VLQ est le dernier octet codant l'entier, mais si $A = 1$, alors au moins un autre octet VLQ suit. Quant à B_n , il s'agit d'un nombre codé sur sept bits, n étant sa position dans le code de l'entier. Une fois les octets VLQ rassemblés, le LSB est placé à la position $n = 0$.

Pour mieux comprendre, prenons l'exemple du nombre 137, et représentons-le d'abord en binaire, soit $137 = 10001001$. Ensuite, séparons ce nombre binaire en parties de sept bits (en commençant par le LSB), ce qui revient à le représenter en base-128, puisque $2^7 = 128$; nous obtenons $137 = 000001\ 0001001$. Prenons à présent les sept derniers bits, afin de déduire l'octet le moins significatif, qui sera donc précédé d'un 0, soit dans notre cas **00001001**. Pour tous les autres groupes de sept bits, il nous suffit d'indiquer que le MSB est égal à 1, ce qui nous donne au final le code du nombre 137 en VLQ, avec en gras les MSB que nous avons ajoutés pour chaque octet — ici, il n'y en a qu'un, donc l'entier 137 est codé sur deux octets VLQ :

$$137 = \mathbf{1}000\ 0001\ \mathbf{0}000\ 1001$$

2.2.3. Les trois types d'évènements d'un bloc de piste

Comme nous l'avons déjà évoqué, la variable <event> accepte trois valeurs : « MIDI », « SysEx » ou « Meta ». Dans tous les cas, le dernier évènement d'un bloc de piste doit être l'évènement méta « *End of Track* » (« fin de piste »). Tous les évènements n'ont pas la même taille, et utilisent le nombre d'octets nécessaire à leur codage. Le premier octet d'un évènement, le « *status byte* », ou « octet de statut », permet d'identifier le type d'évènement parmi les trois qui sont possibles.

```
<event> = <MIDI> ou <SysEx> ou <meta>
<évènement> = <MIDI> ou <Système Exclusif> ou <méta>
```

Figure 8: Les trois types d'évènements que l'on peut trouver dans un bloc de piste

a. Les **MIDI Channel Events** (0x8n - 0xE_n)

Un *MIDI Channel Event* transporte une information de contrôle musical, comme par exemple le déclenchement ou l'arrêt d'une note, ou encore l'ajustement de la *Modulation* ou de l'*Expression*. Comme sa dénomination l'indique, un *MIDI Channel Event* est assigné à un canal MIDI, qui est indiqué par la lettre « n » dans son octet de <type>. Tous les *MIDI Channel Events* se présentent sous la forme suivante :

```
<MIDI Channel Event> = <Δt><type><MIDI channel><parameter 1><parameter 2>
<Évènement de canal MIDI> = <Δt><type><canal MIDI><paramètre 1><paramètre 2>
```

Figure 9: Syntaxe d'un « MIDI Channel Event »

Les *MIDI Channel Events* sont au nombre de sept : *Note Off*, *Note On*, *Note Aftertouch*, *Controller*, *Program Change*, *Channel Aftertouch*, et *Pitch Bend*. Nous n'allons pas nous intéresser à tous ces types de *MIDI Channel Events*, et nous ne retiendrons dans le cadre de ce mémoire que les

événements *Note Off* (type 8), *Note On* (type 9) et *Controller* (type 11). En ce qui concerne la terminologie, nous parlerons indifféremment d'évènement ou de message MIDI pour désigner un *MIDI Channel Event*. Un message *Note Off*, donc, permet de signaler le relâchement d'une touche, autrement dit l'arrêt d'une note, tandis qu'un message *Note On* indique qu'une note est jouée. En-dehors de leur type, les deux paramètres de ces événements sont identiques : le premier renseigne le numéro de la note qui est jouée, c'est-à-dire sa hauteur tonale, tandis que le second rend compte de la vélocité, c'est-à-dire de la « force » avec laquelle la note a été jouée. Dans le cas d'un événement *Note Off*, la vélocité est en général ignorée, bien qu'elle puisse servir à ajuster la pente du *release* (ou « relâchement ») de certains instruments. Quant à l'évènement *Controller*, il signale un changement d'état de l'un des cent vingt-huit contrôles continus qui modifient les attributs d'un canal MIDI, comme son volume, son panoramique, sa modulation, son expression, etc.⁵⁰

b. Les événements SysEx (0xF0 et 0xF7)

Dans un SMF, les événements SysEx sont utilisés pour contrôler du matériel MIDI physique ou logiciel, qui pour fonctionner, nécessite des octets spéciaux définis par leur constructeur. Ainsi, chaque événement SysEx débute par un identifiant (ID), qui indique à quel matériel le message est destiné : tous les membres d'un même réseau ou d'un même système le reçoivent, mais si l'ID reçu ne correspond pas à l'ID spécifié par leur constructeur, ils ignorent tout simplement le message. Il y a trois types d'évènements SysEx : (i) le *Normal SysEx Event*, qui permet de transmettre des données encapsulées dans un seul message, (ii) le *Divided SysEx Event*, qui répartit les données à transmettre sur plusieurs événements, et enfin (iii) le *Authorization SysEx Event*, qui fonctionne comme un péage : il autorise ou non la transmission de messages MIDI spécifiques, comme par exemple des messages de *MIDI Time*

⁵⁰ Une liste des contrôles continus les plus utilisés dans le cadre de la composition de musique de film est donnée en *Annexe 3*.

Code (MTC). Pour résumer, les événements SysEx ne sont pas assignés à un canal MIDI particulier, mais sont destinés à contrôler un élément matériel dans son ensemble, qu'il soit physique ou logiciel.

c. Les événements *méta* (0xFF)

Pour finir, les « *Meta Events* », ou « événements méta », désignent tous les événements qui ne sont pas envoyés ou reçus par le biais d'un port MIDI spécifique. Pour le dire autrement, les événements méta correspondent à des événements spéciaux ; notons qu'ils utilisent le statut 0xFF, qui dans un flux de données MIDI correspond à un *System Reset Message*. Cependant, comme un tel message ne présente aucune utilité dans le cadre d'un SMF, leur statut 0xFF peut être utilisé pour coder les événements méta.

Tous les *Meta Events* débutent donc par *FF*, suivis par un octet définissant son <type>, dont la valeur est comprise entre 0 et 255. Vient ensuite la longueur <length> des données sous la forme d'une quantité de longueur variable, puis enfin les données elles-mêmes. Le fait d'utiliser une quantité de longueur variable pour coder la longueur des données <length> signifie que le champ des données d'un événement méta peut être supérieur à 128 octets. Notons que les types d'événements méta allant de 1 (0x01) à 15 (0x0F) inclus sont réservés à des événements textuels. La logique d'ouverture de la norme MIDI et de rétro-compatibilité pour les *Meta Events* est la même que celle que nous avons déjà détaillée par rapport à la définition de la syntaxe des blocs. Ainsi, leur syntaxe est la suivante :

```
<meta-event> = FF<type><length><bytes>
```

```
<événement Méta> = FF<type><longueur><données>
```

Figure 10 : Syntaxe d'un événement méta

Il faut noter que les évènements méta qui interviennent dans une piste au même moment peuvent intervenir dans un ordre aléatoire. Ainsi, si des *Meta Events* sont essentiels, ou doivent être connus « rapidement » (comme une mention relative aux droits d'auteur, ou encore à un numéro de séquence, un nom de piste, etc.), alors ces derniers doivent être placés au temps 0. Nous allons voir que dans le cadre de la problématique qui nous occupe, certains évènements méta sont beaucoup plus utiles que d'autres. Avant de les détailler, il nous faut préciser que la description de chaque évènement méta est régie par un certain nombre de conventions pour en préciser les paramètres : les caractères « *FF* » que l'on trouve en tête d'évènement, mais aussi le type et la longueur de chaque évènement qui ne sont pas codés en VLQ sont directement donnés en hexadécimal. Une notation comme « *dd* » ou « *se* », c'est-à-dire deux lettres minuscules, est en fait un moyen mnémotechnique pour désigner des valeurs de 8-bits. Logiquement, quatre lettres minuscules consécutives se réfèrent à des valeurs de 16-bits, six à 24-bits, etc., avec à chaque fois le MSB placé en premier.

Enfin, même si la plupart des évènements méta sont optionnels, quelques-uns sont obligatoires, ce qui signifie que si leur valeur n'est pas renseignée dans le SMF, elle est fixée par défaut par rapport aux recommandations de la norme MIDI. Ces évènements méta obligatoires sont listés ci-dessous :

End of Track <FF 2F 00> : cet évènement de « fin de piste » doit obligatoirement être le dernier évènement présent dans un bloc de piste, et ne présenter une seule et unique occurrence par piste.

Tempo <FF 51 03 tt tt tt> : <tt tt tt> est une valeur de 24 bits codant le *tempo* en microsecondes par noire. Musicalement, il est d'usage de spécifier le *tempo* en battements par unité de mesure du temps — le plus souvent en battements par minute (BPM) —, mais la spécification MIDI propose l'inverse, c'est-à-dire de définir le *tempo* en

temps par battement. Cela permet d'assurer une synchronisation plus précise sur le long-terme avec un protocole de synchronisation basé sur le temps comme le *MIDI Time Code* ou le TC SMPTE. Si aucun tempo n'est spécifié en tête du fichier MIDI, il sera fixé par défaut à 120 BPM, soit $5 \times 10^6 \mu s$ par noire.

Time Signature <FF 04 nn dd cc bb> : cet évènement de signature rythmique nécessite quatre valeurs : <nn> nous donne le numérateur de la signature rythmique tel qu'il est noté, <dd> indique le dénominateur de la signature rythmique comme une puissance de 2 négative⁵¹, où « 2 » représente une noire, « 3 » une croche, etc. Ensuite, <cc> nous renseigne sur le nombre de « *MIDI Clocks* » entre les clics du métronome, et enfin <bb> spécifie le nombre de triples croches notées dans une noire MIDI ; la valeur usuelle de <bb> est 8, mais certains logiciels permettent à l'utilisateur de donner une autre valeur à ce que le protocole MIDI note et interprète comme une triple-croche. Par exemple, une signature rythmique de 4/4 avec un métronome cliquant toutes les noires sera codée comme suit : <FF 58 04 04 04 02 18 08>. Enfin, si aucune signature rythmique n'est spécifiée en tête du SMF, elle est par défaut fixée à 4/4.

Key Signature <FF 59 02 sf mi> : cet évènement décrit la signature harmonique, ou l'armure du SMF. Il est codé grâce à deux données : <sf> et <mi>. La donnée <sf>, si elle est négative, indique le nombre de bémols à l'armure, et renseigne logiquement le nombre de dièses si elle est positive. Le paramètre <mi>, quant à lui, indique si la tonalité est majeure (0) ou mineure (1). Si le morceau débute dans les tonalités de Do majeur ou de La mineur, la valeur de <sf> est nulle.

51 Une puissance de 2 négative est l'inverse du résultat d'une puissance de 2 positive, soit $2^{-n} = 1/2^n$. Par exemple, $2^{-3} = 1/2^3 = 1/8$. Dans notre cas, on code la puissance de 2 négative qui correspond à une subdivision de la ronde. Une noire est un quart de ronde, donc pour une noire, <dd> = 2, puisque $2^{-2} = 1/2^2 = 1/4$.

Pour conclure sur tous les types d'évènements MIDI, nous devons relever que si l'octet de statut d'un évènement MIDI est inférieur à 128 (x80), cela implique que le « *Running Status* » est en action et que cet octet, dont la valeur est inférieure à 128, est en réalité un premier octet de données. En effet, il n'est pas rare de transmettre des séries d'évènements consécutifs d'un même type. Par exemple, quand un accord est joué sur un clavier, une demi-douzaine d'évènements *Note On* peuvent être générés, suivis d'une demi-douzaine d'évènements *Note Off*. Le protocole MIDI étant sériel, cela peut poser quelques problèmes de synchronisation : quand le *Running Status* est en fonctionnement, l'octet de statut d'un évènement n'est transmis que s'il diffère de l'évènement qui le précède sur un même canal. Bien évidemment, il demeure nécessaire, voire même obligatoire, de préciser le statut du premier évènement de chaque bloc de piste. Cela signifie que lorsque le *Running Status* est en action, le statut du message est déduit du message précédent. Par définition, cela n'est possible que pour des évènements assignés à un ou des canaux MIDI : par conséquent, les évènements SysEx et méta interrompent systématiquement le *Running Status*.

Maintenant que la structure d'un SMF a été détaillée, il convient de nous intéresser à l'utilisation du MIDI dans une STAN. Autrement dit, il nous faut plus précisément aborder l'utilisation du protocole MIDI, qui permet de gérer des flux de données qui sont transmises en temps réel. Une fois que le SMF a été importé dans la STAN, il est lu et interprété comme s'il était joué en direct, comme s'il s'agissait d'une performance musicale en temps réel.

3. Utilisation du MIDI dans une STAN

Un « séquenceur » est un logiciel dédié à la musique, capable d'enregistrer et d'exécuter une séquence de commandes. Il ne produit pas de son par lui-même, mais sert à automatiser l'exécution d'une séquence

musicale par le biais de messages numériques, dont le format le plus répandu est bien évidemment le MIDI. Par abus de langage, nous désignons parfois une STAN par le terme « séquenceur », mais nous allons voir qu'il ne s'agit que de l'une des nombreuses fonctionnalités qu'elle peut proposer.

3.1. Définitions et fonctionnalités

Historiquement, nous pouvons considérer que les premiers séquenceurs sont des instruments de musique mécaniques comme la boîte à musique ou l'orgue de barbarie. Les premiers modules analogiques de séquenceurs sont apparus dans les années 1970 pour subvenir aux besoins de la musique électronique. A leurs balbutiements, ils ne sont pas encore capables de gérer de larges et longues séquences de notes : en 1971, par exemple, le *EMS Synthi AKS* permet d'enregistrer des séquences de deux cents cinquante-six notes. Au début des années 1980, l'arrivée du numérique et l'invention du protocole MIDI, dans un but de standardisation de la communication entre les machines et les instruments d'une part, et d'élargissement du nombre des paramètres contrôlables et automatisables par l'utilisateur d'autre part, ont permis de gérer des séquences de notes plus longues et plus complexes. La puissance de calcul et de stockage grandissant des stations de travail, autrement dit des ordinateurs, a enfin permis aux séquenceurs logiciels de gérer des séquences sans limite de taille ou de longueur.

L'avènement de la MAO dans les années 1970 rend possible le développement de nombreux outils logiciels avec des fonctionnalités de plus en plus complètes. Aujourd'hui, nous rassemblons l'ensemble de ces logiciels dédiés à la composition musicale et au traitement de l'audio sous l'acronyme « STAN » (« station de travail audionumérique », ou *DAW* pour « *Digital Audio Workstation* »). Ces STAN sont parfois appelées « séquenceurs », mais il s'agit d'un abus de langage. En effet, une STAN est un logiciel plus complet qui intègre, entre autres fonctionnalités, un séquenceur. Par exemple, *ProTools*

(*Avid*) est à la fois un séquenceur, un échantillonneur, un processeur d'effets et un contrôleur MIDI, tandis que *Cubase Pro* (*Steinberg*) possède toutes ces fonctionnalités, auxquelles s'ajoutent un éditeur de partitions.

Un échantillonneur, ou « *sampler* » en anglais, est un instrument de musique physique ou numérique capable d'enregistrer des échantillons (des « *samples* ») et de les reproduire en leur appliquant ou non des traitements. Ces échantillons peuvent être issus de percussions, d'instruments solistes, de groupes d'instruments, d'un orchestre, etc. En général, un échantillonneur est commandé par le biais d'un clavier-maître, qui permet de déclencher la lecture d'un ou de plusieurs échantillon(s). On utilise aujourd'hui principalement des échantillonneurs numériques, sous la forme de logiciels — on parle dans ce cas de fonctionnement en « *standalone* » — ou de *plug-ins*⁵². Les plus répandus et les plus utilisés aujourd'hui sont *Kontakt* (*Native Instruments*), *Halion* (*Steinberg*), *UVI Workstation* (*UVI*) ou encore *Play* (*Eastwest*). Nous verrons plus tard que l'échantillonneur est un outil central et essentiel dans la chaîne de production d'une musique de film, puisqu'il est notamment nécessaire à la création de maquettes.

Le second élément présent dans certaines STAN et qui pourrait nous intéresser est l'éditeur de partitions. Il faut évidemment différencier les éditeurs de partition qui sont intégrés dans des STAN comme *Cubase Pro* ou *Logic Pro X*, et les logiciels exclusivement dédiés à la notation et à la gravure musicales, comme *Finale* (*MakeMusic*) ou *Sibelius* (*Avid*). Les premiers sont en général assez limités, et donc très rarement utilisés pour éditer et graver des partitions définitives. En effet, les logiciels entièrement destinés à l'édition de partitions proposent beaucoup plus de symboles, de notations et de

52 Un *plug-in* est un module d'extension qui va compléter un logiciel hôte, ici notre STAN, pour lui apporter de nouvelles fonctionnalités. Un *plug-in* permet typiquement d'intégrer des instruments virtuels ou des effets audio supplémentaires. Dans la suite de ce mémoire, nous avons fait le choix de ne pas traduire ce terme, qui est utilisé tel quel par les utilisateurs de STAN, qu'ils soient francophones ou anglophones, les Québécois préfèrent parler de « *plugiciel* ».

fonctionnalités, et sont donc préférés aux premiers pour éditer les conducteurs et les parties séparées qui seront utilisés en séance d'enregistrement.

3.2. Les différences entre un flux de données et un fichier MIDI

Le MIDI est omniprésent dans une STAN en tant que protocole de communication, c'est-à-dire en tant que flux de données traitées en temps réel, et non en tant que structure de fichier. Ainsi, le SMF est utilisé pour exporter et importer des données depuis ou dans la STAN, mais une réorganisation des données est nécessaire pour leur exploitation. Pour clarifier ce qui distingue un fichier d'un flux de données MIDI, nous devons nous rappeler que dans un SMF, un Δt est associé à chaque évènement, tandis que dans un flux de données MIDI, les évènements ont lieu au moment où ils sont reçus, ou joués. De plus, les *System Common Messages* et les *System Real Time Messages* ne sont pas utiles et n'apparaissent donc pas dans un SMF, car ils ne présentent aucune utilité : ils sont remplacés par une séquence de fuite (« *Escape Sequence* ») lorsqu'une fonctionnalité qu'ils peuvent transporter doit être convoquée de manière occasionnelle. Enfin, dans un fichier MIDI, l'identifiant du *System Reset Status Byte* est utilisé pour identifier un évènement méta, d'où le fait qu'on ne trouve de *Meta Events* que dans les SMF. Nous allons voir qu'une STAN, de par ses fonctionnalités de séquenceur, nous permet de « simuler » une performance musicale jouée en direct, afin de piloter, grâce au protocole MIDI, des instruments virtuels. Toutes les informations globales contenues dans le premier bloc de piste d'un SMF de *Format 1* sont réparties sur des pistes « spéciales », dédiées à les accueillir dans la STAN, qui permet donc la création de pistes dédiées à la gestion du *tempo*, de la signature rythmique et de l'armure sur l'ensemble d'une session de travail.

3.3. Les pistes « MIDI » et les pistes « Instrument »

Pour mieux comprendre l'utilisation et le fonctionnement du MIDI dans une STAN, nous nous proposons de l'envisager du point de vue de l'utilisateur, c'est-à-dire de l'approcher par la pratique. Pour commencer, il est nécessaire de distinguer les pistes « Instrument » des pistes « MIDI ».

Une piste *Instrument* accueille, comme sa dénomination le laisse entendre, un « instrument virtuel », qui est en général associé à un échantillonneur logiciel. Un instrument virtuel est un instrument de musique logiciel utilisé en MAO pour produire des sons « électroniques », ou reproduire des sons d'instruments réels ; nous parlerons aussi de « synthétiseurs virtuels ». Il existe trois types d'instruments virtuels : les échantillonneurs, les synthétiseurs, et enfin les boîtes à rythmes, qui sont basées sur des échantillons et/ou de la synthèse d'instruments percussifs — ou autres, d'ailleurs —, mais qui ont surtout la particularité de proposer des motifs rythmiques ; les boîtes à rythmes embarquent en général un séquenceur. Nous désignerons donc par « instrument virtuel » tout instrument de musique logiciel piloté par commande numérique — sous-entendu en MIDI —, intégré à notre STAN sous la forme d'un *plug-in*, ou fonctionnant indépendamment.

Une piste *MIDI*, quant à elle, pilote des pistes *Instrument* ou des périphériques MIDI externes. Des données MIDI sont enregistrées sur ces pistes *MIDI*, qui sont renvoyées en temps réel aux pistes *Instrument* durant la lecture. C'est la partie « séquenceur » de la STAN qui est mobilisée, puisqu'on enregistre des séquences de notes plus ou moins complexes, associées à des contrôles continus, ou « *Control Changes* » (CC). L'utilisation des *Control Changes* sera détaillée ensuite : nous retiendrons pour le moment qu'ils modifient les paramètres d'un instrument ou d'un matériel MIDI physique ou virtuel, ce qui permet à la fois de les contrôler, de les automatiser, et de les personnaliser.

B. Les instruments virtuels et leur utilisation

Nous avons précédemment défini ce que nous entendions par « instrument virtuel », et avons établi que les méthodes de travail des compositeurs de musique à l'image avaient beaucoup évolué au fil des années : nous sommes passés d'ébauches composées en général au piano et écrites sur une poignée de portées à des maquettes MIDI réalistes et pré-orchestrées. Les progrès en matière d'échantillonnage et la prolifération des instruments virtuels ne sont certainement pas étrangers à cette évolution, et ils sont aujourd'hui au cœur de l'espace et des méthodes de travail des compositeurs, notamment de musique de film.

1. Les différentes familles d'instruments virtuels

Nous avons déjà vu qu'il existait trois types d'instruments virtuels : (i) les échantillonneurs, qui compilent des échantillons, (ii) les synthétiseurs, qui fonctionnent uniquement grâce à des algorithmes de synthèse sonore, et enfin (iii) les boîtes à rythmes, qui peuvent être basées sur l'échantillonnage et/ou la synthèse sonore, mais qui présentent la particularité de proposer des motifs rythmiques et/ou un séquenceur intégré. Depuis quelques années, certains éditeurs développent même des instruments virtuels hybrides. Ces instruments, à l'instar du *Falcon*, commercialisé par UVI en 2015, permettent de combiner à la fois des échantillons et des algorithmes de synthèse sonore. De ce fait, ils sont davantage tournés vers la création sonore ou le *sound design*⁵³ que vers l'émulation de célèbres synthétiseurs analogiques, ou de la

53 Dans l'industrie cinématographique, le *sound designer* ou « concepteur sonore » se confond souvent avec le monteur son. En général, on entend par *design sonore*, un travail de création autour de matières sonores diverses et variées dans le but de créer des sonorités originales et inédites. Le travail de Ben Burtt, qui a notamment sonorisé *Star Wars* et qui revendique le terme de « *sound designer* », nous en donne de nombreux exemples, comme l'utilisation du barrissement d'un éléphant pour créer l'empreinte sonore des *chasseurs TIE*.

reproduction d'instruments acoustiques. Ainsi, les instruments virtuels réunissent et accumulent au sein d'une STAN une panoplie d'instruments et de sonorités infiniment étendue. Le compositeur dispose donc dans son studio d'un orchestre symphonique, de chœurs, de pianos, de guitares, de batteries, de percussions diverses et variées, d'émulations de synthétiseurs analogiques ou numériques, et d'autres « instruments » innovants et originaux. Il peut les jouer, les piloter, et contrôler [tous] leurs paramètres grâce à un clavier maître et à toutes sortes de contrôleurs MIDI.

Au sein de son studio, le compositeur est donc en mesure de produire des maquettes, dont certaines pistes seront réenregistrées par de véritables instruments, ce qui ne l'empêchera pas d'en conserver d'autres dans le mixage final de la musique, comme des synthétiseurs ou des percussions. Dans le cas d'une musique de film mêlant un orchestre symphonique, quelques instruments acoustiques additionnels, comme une batterie des guitares, ainsi que quelques synthétiseurs virtuels ou physiques, nous pourrions par exemple enregistrer l'orchestre symphonique, la batterie et les guitares, et conserver les synthétiseurs virtuels tels quels, ou encore, notamment pour des raisons budgétaires, choisir d'enregistrer la batterie, les guitares et les instruments solistes de l'orchestre, et conserver les synthétiseurs et les autres pupitres de l'orchestre dans leur état virtuel.

Comme le souligne Richard Davis, « la technologie nous a permis de créer des partitions de bonne qualité sonore pour un petit budget »⁵⁴, c'est-à-dire des maquettes relativement vraisemblables, qui peuvent être conservées dans la version définitive de la bande son du film. C'est notamment le cas de films à petit budget, de téléfilms ou de certaines séries télévisées, qui n'ont pas les moyens d'enregistrer un orchestre symphonique. Quoiqu'il en soit, une maquette, même très aboutie, ne remplacera jamais un orchestre composé de musiciens de haut niveau, enregistré et produit dans une acoustique traitée et appropriée. Dans la mesure du possible, il est donc souhaitable d'enregistrer

⁵⁴ « Technology enables us to make great sounding scores for a small budget. » — Richard Davis, *op. cit.*, p. 76.

de vrais musiciens, même si cela n'est pas toujours possible. Nous pourrions opposer à cela que les progrès en matière d'échantillonnage suivent une trajectoire exponentielle, et que les développeurs de banques d'instruments virtuels ont significativement évolué ces dernières années. Par conséquent, la majorité du public ne fait plus la différence entre un véritable orchestre et un orchestre échantillonné. Cela peut nous amener à penser que les banques orchestrales haut de gamme commercialisées par *Orchestral Tools* ou *Spitfire* peuvent rivaliser avec la majorité des orchestres symphoniques. Cependant, même si ces banques offrent de nombreux paramètres pour jouer sur l'interprétation et la musicalité de chaque instrument, il demeure néanmoins difficile et chronophage d'imiter « à la perfection » le jeu d'un véritable musicien, si tenté qu'il soit réellement possible d'atteindre cette « perfection ». Pour résumer, il faut distinguer les instruments virtuels qui resteront à l'état de maquette, c'est-à-dire que nous ne retrouverons pas dans la version mixée de la musique du film, et les instruments virtuels qui seront utilisés tels quels.

2. La gestion et l'utilisation d'un studio hybride

Le studio d'un compositeur de musique à l'image est aujourd'hui hybride : l'audio numérique cohabite avec l'analogique et le physique avec le virtuel. C'est aussi le cas de la plupart des studios d'enregistrement, bien que la partie réservée aux instruments virtuels y soit bien moindre. De plus, la production musicale est échelonnée en plusieurs phases, ce qui induit que certains équipements sont utilisés au moment de l'enregistrement et d'autres au moment du mixage ou du *mastering* ; ce n'est pas le cas d'un studio de composition, qui doit être en mesure de tout réaliser simultanément, sans perte de temps. Si l'essentiel d'une maquette est créé grâce à des instruments virtuels, il n'est pas rare que le compositeur enregistre, dès la phase de composition, un certain nombre d'instruments, comme un piano, une batterie, des percussions, des guitares, des synthétiseurs analogiques, etc. Par ailleurs,

le compositeur travaille déjà sur la production, c'est-à-dire sur la couleur et l'esthétique sonores de sa bande originale, dès la composition, ce qui peut l'amener à utiliser des effets externes à sa STAN, analogiques ou numériques, comme des réverbérations, des compresseurs, etc. Toutes ces possibilités doivent être centralisées et pré-configurées autant que possible au sein de la STAN. Autrement dit, la STAN est à la fois le cœur et la colonne vertébrale du studio de composition moderne.

2.1. La cohabitation entre le physique et le virtuel

Au risque de nous répéter, la principale contrainte rencontrée par les compositeurs de musique à l'image est sans aucun doute la gestion du temps. Nous avons vu à quel point la technologie est au centre de leur processus de travail, et bien qu'elle soit d'une grande aide, elle ne doit en aucun cas se muer en un obstacle ou une barrière à la créativité et à l'imagination. C'est d'ailleurs pour cette raison que les compositeurs ont de plus en plus eu recours à des assistants multi-tâches au fur et à mesure du développement et de la prolifération de la MAO : un studio doit fonctionner, sa mécanique doit être huilée, et toutes les combinaisons de liaison ou de communication entre tous les appareils et toutes les machines qui le composent envisagées. Lorsqu'il s'assoit devant sa station de travail, le compositeur ne doit penser qu'à écrire, qu'à inventer ; il doit pouvoir être en mesure d'expérimenter sans contrainte. C'est pour cela que tout doit être prêt à être utilisé en permanence.

Nous ne développerons pas ici l'élaboration d'un réseau de câblage complexe, mais nous nous contenterons de retenir que toutes les sources peuvent entrer ou sortir de la STAN d'une manière ou d'une autre. Qu'il s'agisse d'effets externes, de machines analogiques ou numériques, d'ordinateurs, ou encore de microphones, ils doivent tous être prêts à l'emploi. Il n'y a pas grand-chose à dire de plus sur la gestion du matériel physique, si ce n'est qu'il est préférable que les instruments régulièrement joués soient

équipés en permanence de microphones, afin de pouvoir être enregistrés à loisir. Par contre, nous ne pouvons pas faire ici l'impasse sur la gestion des banques de son.

Une *banque de son*, dans le domaine de la musique, est une bibliothèque contenant un certain nombre d'instruments virtuels. Les éditeurs de ces banques sont en général spécialisés dans un domaine musical ou un certain type d'instruments, et développent toute une gamme de banques relatives à ce domaine. Par exemple, *Spitfire*⁵⁵ ou *Orchestral Tools*⁵⁶ sont spécialisés dans les instruments qui composent un orchestre symphonique, et proposent principalement des banques de cordes, de bois, de cuivres, de percussions, etc. *Spectrasonics*⁵⁷ s'oriente plutôt vers les synthétiseurs et les sonorités électroniques, tandis que *Synthogy*⁵⁸ s'est spécialisé dans les banques de pianos. *Native Instruments*⁵⁹ est quant à lui beaucoup plus généraliste et éclectique, puisque l'éditeur propose aussi bien des banques de percussions cubaines, de guitares acoustiques, de sections de cuivres « pop », que de synthétiseurs en tous genres. Il existe des dizaines d'éditeurs qui proposent chacun des dizaines de banques de son : chaque compositeur a ses préférences pour tel ou tel éditeur et pour tel ou tel type d'instrument, mais en général, il se construit progressivement une palette d'instruments virtuels provenant de plusieurs d'entre-eux. Cela nécessite des espaces de stockage relativement conséquents, puisque l'accumulation de ces banques de son peut représenter plusieurs téraoctets de données.

C'est notamment pour pallier cette problématique de stockage — et donc d'exploitation — qu'un studio de composition est en général construit autour de plusieurs ordinateurs : un « Maître », sur lequel est exécuté la STAN, et un ou plusieurs « Esclave(s) », qui permettent de stocker et de gérer les banques de

55 <https://www.spitfireaudio.com> (dernière consultation le 30/04/2018).

56 <http://www.orchestraltools.com> (dernière consultation le 30/04/2018).

57 <https://www.spectrasonics.net> (dernière consultation le 30/04/2018).

58 <http://synthogy.com> (dernière consultation le 30/04/2018).

59 <https://www.native-instruments.com/fr/> (dernière consultation le 01/05/2018).

son, mais aussi divers effets numériques, comme par exemple des réverbérations. Nous verrons un peu plus tard comment le logiciel *Vienna Ensemble PRO* facilite la communication entre le *maître* et ses *esclaves*.

Afin de parvenir à atteindre tous ces objectifs et de développer un studio hybride le plus optimisé possible, la majorité des compositeurs travaillent avec des « *templates* » au sein de leur STAN, c'est-à-dire avec des « modèles » de session. Celles-ci permettent d'organiser la cohabitation du numérique et de l'analogique, du virtuel et du physique, ainsi que l'accessibilité de l'ensemble des outils et des fonctionnalités qu'ils proposent. La conception d'une *template*⁶⁰ est très personnelle et propre à chacun, mais il est aujourd'hui essentiel pour un compositeur d'y réfléchir, puisque cette dernière va conditionner ses méthodes de travail. En toute logique, il est plus productif et plus efficace de conditionner notre *template* par rapport à nos méthodes de travail, plutôt que de laisser une *template*, c'est-à-dire l'architecture et l'organisation de notre studio hybride, conditionner notre approche de la composition et du processus créatif.

2.2. La construction de *template(s)*

La construction et l'utilisation de *templates* n'est pas propre à la composition musicale [à l'image]. Cette pratique est même assez répandue dans le monde de l'audiovisuel. Une *template* est un modèle, un gabarit, ou encore un patron, c'est-à-dire une base de travail. Dans le cas de la musique, il s'agit d'une session de travail créée dans notre STAN, et dans laquelle un maximum d'éléments ont été préparés. Il existe plusieurs manières d'aborder la conception et la construction de *templates*, qui peuvent se résumer en deux possibilités : (i) créer une unique *template* qui contient tous les éléments qui

60 Pour les mêmes raisons que nous avons déjà évoquées par rapport à un certain nombre d'anglicismes, nous emploierons dans la suite de ce mémoire le terme de « *template* » au lieu de celui de « session modèle » ou de « modèle ».

sont à notre disposition, ou (ii) créer plusieurs *templates* moins imposantes, et les utiliser telles quelles ou les combiner suivant nos besoins.

Dans le premier cas, notre *template* contient tous les instruments virtuels dont nous pourrions avoir besoin, ainsi que toutes les pistes nécessaires à gérer les signaux provenant de nos diverses machines physiques. De plus, tous les chemins possibles des signaux audio, MIDI, ou autres, à l'intérieur et en-dehors de notre STAN, doivent avoir été configurés afin de pouvoir communiquer, enregistrer et exporter dans les meilleures conditions possibles ; il s'agit du « *routing* » de notre installation. Il est également possible de préparer tous les effets dont nous pourrions avoir besoin et de les prérégler. Cela consiste notamment à mettre en place des réverbérations ou encore des tranches de *mastering*. Le but d'une *template* est de ne pas avoir à reconfigurer en permanence notre installation, et surtout de ne pas avoir à répéter les mêmes tâches au début de chaque nouveau projet, comme par exemple un réglage d'égalisation sur notre piste de premiers violons.

La seconde méthode consiste à imaginer plusieurs *templates* suivant différentes applications. Par exemple, nous pouvons créer des *templates* « quatuor à cordes », « orchestre symphonique », « *big band* », ou encore « groupe de rock ». Paul Gilreath explique qu'il utilise « une cinquantaine de *templates* basiques, et que chacune d'elles est dédiée à un genre ou à une situation spécifiques »⁶¹. On peut évidemment combiner ces différentes *templates* de base pour en créer une plus adaptée à un type de projet, voire même en fabriquer une nouvelle à chaque nouveau projet. Il est tout à fait envisageable de faire évoluer une *template* pendant le processus de création, mais c'est une opération qui peut parfois s'avérer périlleuse, puisqu'une mauvaise manipulation pourrait compromettre l'ensemble de la session.

61 « I have about fifty or more basic templates that I use, and each one is specific to a writing situation or genre. » — Paul Gilreath, *The Guide to MIDI Orchestration*, Burlington, Focal Press, 1996, 4^{ème} éd. (2010), p. 207.

Nous retiendrons donc qu'une *template* est un modèle de session qui permet au compositeur d'avoir sous les yeux et à portée de main — ou de clavier maître — l'ensemble de sa palette sonore, ou du moins la palette sonore qu'il juge nécessaire et pertinente pour le projet qu'il est en train de mettre en musique. En outre, la *template* concentre dans une STAN l'ensemble du studio virtuel du compositeur : sa mise en place peut donc s'avérer longue et nécessiter de la minutie, mais au final, elle offre un gain de temps et de productivité considérable.

Néanmoins, la principale difficulté dans l'élaboration d'une *template* réside dans sa calibration, puisque sa taille et son exhaustivité, en termes d'instruments disponibles, sont étroitement liées à la puissance et aux capacités de l'ordinateur qui héberge la STAN, ou les banques de son dans le cas d'une configuration avec plusieurs ordinateurs. Les instruments virtuels étant particulièrement gourmands en mémoire vive (RAM), cette dernière peut rapidement être saturée. C'est notamment pour cette raison qu'il est souvent préférable de soulager l'ordinateur *maître* de la gestion des banques de son, et de les déporter vers un ou plusieurs ordinateurs *esclaves*. Le logiciel *Vienna Ensemble PRO* est alors un allié précieux, et un élément important de la configuration d'un studio de composition.

2.3. Vienna Ensemble PRO (VSL)

Le logiciel *Vienna Ensemble PRO* (VEP) est développé par *Vienna Symphonic Library* (VSL)⁶². Il s'agit du premier réseau local tout-en-un qui allie à la fois MIDI et audio, et qui fonctionne aussi bien sur les systèmes d'exploitation développés par *Apple* et par *Microsoft* ; il s'assimile conjointement à un mixeur virtuel et à un hôte de *plug-ins*. VEP permet de charger des instances de lecteur d'échantillons que nous avons déjà évoqués,

62 VSL, ou « *Vienna Symphonic Library* » : <https://www.vsl.co.at/en> (dernière consultation le 30/04/2018).

tels que *Kontakt* (Native Instruments), *Play* (Eastwest) ou encore *UVI Workstation* (UVI), mais aussi des synthétiseurs virtuels comme *Omnisphere* (Spectrasonics), *Massive* (Native Instruments), ou *Falcon* (UVI). Le mixeur permet en outre d'utiliser tous les *plug-ins* que nous pouvons trouver sur le marché aux formats *AU*, *VST*, *VST3*, *AAX Native* et *MAS*⁶³. Le principe de *VEP* est de construire un réseau local constitué d'au moins une machine *maître*, qui peut avoir ou non des *esclaves*. Dans tous les cas, les ordinateurs qui constituent le réseau local sont simplement connectés entre eux par un câble *ethernet*.

Dans les faits, *VEP* est constitué de deux applicatifs : *Vienna Ensemble PRO*, qui fonctionne en *standalone*, et que nous n'aborderons donc pas ici, et *Vienna Ensemble PRO Server*. Pour être tout-à-fait précis, il s'agit de créer des instances de *VEP* à l'intérieur de *VEP Server*, et c'est dans ces instances de *VEP* que nous pourrions ouvrir nos instruments virtuels et les *plug-ins* dont nous avons besoin. Pour le dire autrement, il est possible de créer de nombreuses instances dans *VEP Server* auxquelles nous pourrions nous connecter, et avec lesquelles nous pourrions communiquer *via* le réseau local.

Nous avons expliqué que *VEP* est une solution pour déporter nos banques de son sur un autre ordinateur, afin de soulager notre STAN. À vrai dire, *VEP* offre avant tout la possibilité de gérer nos banques de son en-dehors de notre STAN : son fonctionnement est similaire indépendamment de la machine sur

⁶³ *AU* (*Audio Units*) est l'architecture des *plug-ins* fonctionnant sur le système d'exploitation *Mac OS X*, ou *macOS*. Son concurrent direct est le *VST* (*Virtual Studio Technology*), développé par *Steinberg*, qui est un format de *plug-in* ouvert fonctionnant sur *Windows*, *Mac OS X*, *macOS* ou *Linux* ; le *VST3* est une mise à jour du format *VST*. *Steinberg* a d'ailleurs annoncé sur son site (<https://www.steinberg.fr/fr/home/news/details/article/le-vst-2-touche-a-sa-fin-4734.html>, dernière consultation le 20/05/2018) qu'à partir du mois d'octobre 2018, seule la norme *VST3* continuerait à être développée. L'*AAX Native* est quant à lui un format propriétaire développé par *Avid* et compatible uniquement avec *ProTools*, depuis sa version 10 ; il remplace le format *RTAS* (*Real Time Audio Suite*), jugé insuffisant par *Avid* et définitivement abandonné depuis *ProTools 11*. Enfin, le format *MAS* (*MOTU Audio System*) est le format natif des *plug-ins* de *Digital Performer*.

laquelle il se situe, qu'il s'agisse du *maître* ou d'une machine *esclave*. L'usage d'un *esclave* permet de soulager la STAN au niveau logiciel et physique, mais rien n'empêche *VEP Server* de cohabiter avec notre STAN sur une même machine : il n'induit alors « qu'un » soulagement logiciel.

Au niveau de notre STAN, *VEP* se présente sous la forme d'un *plug-in* par le biais duquel on établit une connexion entre la piste sur laquelle il a été inséré et une des instances créées dans *VEP Server*. Chaque *plug-in* de *VEP* dans notre STAN ne peut se connecter qu'à une seule instance. De plus, il est possible de définir pour chaque instance de *VEP* un nombre indépendant d'entrées et de sorties audio, mais aussi de ports MIDI. De nombreuses possibilités de *routing* et de configuration de session nous sont ainsi accessibles. Nous pouvons utiliser un nombre conséquent d'instruments virtuels, tout en allégeant drastiquement notre système principal dans le cas d'un réseau local comprenant au moins un *esclave*. Notons que *VEP* reçoit et envoie aussi bien des données MIDI que des flux audio.

Dans une configuration standard, des informations MIDI sont enregistrées sur une piste MIDI au sein de notre STAN ; à la lecture, ces données sont envoyées à une instance de *VEP*, qui accueille un instrument virtuel qui est piloté par les données qu'il reçoit. L'instrument est « joué », et la sortie audio de *VEP Server* de cet instrument revient dans une piste audio de notre STAN. *Grosso modo*, deux pistes sont nécessaires pour un même instrument, à savoir une piste MIDI et une piste audio. Bien entendu, si nous avons plusieurs instruments virtuels pour un même instrument, les sorties de ces instruments sont sommées dans un bus qui est envoyé dans une seule piste de notre STAN. En général, ce sont donc nos *stems* qui reviennent en audio de *VEP* vers notre STAN. Par exemple, nous pourrions avoir quatre pistes MIDI pour les *Violons I*, mais une seule piste audio *Violons I*, voire même une seule piste pour les cordes aigües, laquelle serait la sommation des premiers et des seconds violons. Comme *VEP* transporte aussi bien des flux MIDI que des flux audio, il est tout à fait envisageable de déporter nos réverbérations ou

d'autres effets gourmands en calculs pour notre machine principale, mais nous n'entrerons pas ici dans les détails de configurations plus complexes.

Par ailleurs, l'immense avantage de *VEP* est de pouvoir charger une seule fois en RAM tous les éléments dont nous avons [toujours] besoin, puis de changer de session autant de fois que nous le souhaitons sans avoir à les recharger à chaque fois : en d'autres termes, *VEP* offre un gain de temps considérable aux compositeurs qui ont des sessions — ou des *templates* — conséquentes, voire gigantesques. En effet, la majorité des STAN vident le cache de la mémoire vive à la fermeture d'une session. Or, une session qui nécessite de charger en mémoire vive plusieurs dizaines de gigaoctets de données peut prendre quelques dizaines de minutes à être opérationnelle. Pour résumer, *VEP Server* devient rapidement incontournable pour un compositeur de musique à l'image dès lors qu'il écrit pour orchestre, et a besoin de gérer un très grand nombre d'instruments virtuels, sans que cela ne le gêne ou ne le ralentisse dans son travail et son processus créatif.

3. Enregistrement et production d'une maquette MIDI

Maintenant que nous avons cerné les enjeux inhérents à la configuration d'un studio de composition hybride, et que nous avons abordé la gestion des banques de son au niveau macroscopique, il convient de nous attarder sur l'enregistrement et la production d'une maquette MIDI, et plus précisément sur la jouabilité et l'interprétation des instruments virtuels.

Nous avons vu que les éditeurs de banques de son se spécialisent en général dans un domaine ou une famille d'instruments : à titre d'exemple, la banque *Berlin Strings*, développée par l'éditeur *Orchestral Tools*, propose le quintette à cordes d'un orchestre symphonique, avec pour chaque instrument des dizaines d'articulations différentes. *Berlin Strings* est ainsi subdivisée en six « sous-bibliothèques » : *Ensemble*, c'est-à-dire l'orchestre à cordes au

complet, *Violons I*, *Violons II*, *Alti*, *Violoncelles* et *Contrebasses*. Ces sous-bibliothèques contiennent chacune des dizaines d'« instruments », que nous appelons en réalité des « *patches* », qui correspondent à une articulation particulière, c'est-à-dire à un mode de jeu. Par exemple, la sous-bibliothèque *Violons I* contient les *patches* *Violons I Legato*, *Violons I Staccato*, *Violons I Spiccato*, etc. Pour avoir à sa disposition toute la palette et tous les modes de jeu du pupitre des *Violons I*, le compositeur doit donc combiner ces différents *patches* suivant ses besoins. La gestion des articulations et des modes de jeu d'un instrument virtuel est par conséquent au cœur des problématiques techniques et méthodologiques intrinsèques à la composition de musique de film moderne.

Il est essentiel de pouvoir passer d'une articulation à une autre au cours d'un *cue*, ce qui est aujourd'hui possible grâce à deux méthodes : les *keyswitches*⁶⁴ et/ou l'assignation d'une articulation par piste ou par canal MIDI. Après avoir développé ces deux méthodes, nous aborderons la gestion de la dynamique d'un instrument, c'est-à-dire son « volume » — en musique, nous préférons parler de « nuance » —, qui a une influence directe sur son timbre, et enfin, il sera succinctement question du pré-mixage et des enjeux relatifs au maquettage d'un *cue*.

3.1. Gestion des articulations et des modes de jeu

Les *keyswitches* sont des notes qui ne sont pas comprises dans la tessiture de l'instrument qui est joué, et qui permettent de déclencher un changement d'articulation lorsqu'elles sont pressées ; bien évidemment, ces *keyswitches* doivent avoir été préalablement programmés par l'éditeur de l'instrument virtuel concerné. Pour les utiliser, au lieu de nous servir d'un

⁶⁴ Comme « *cue* » ou « *score* », le terme « *keyswitch* » est un anglicisme qui est entré dans le vocabulaire des compositeurs [francophones] qui utilisent des instruments virtuels ; *Steinberg*, dans le manuel de *Cubase Pro 9*, traduit le terme par « *commutateur* », mais dans la suite de ce mémoire, nous préférons tout de même conserver celui de *keyswitch*.

patch qui ne comporte qu'une seule articulation — comme *Violons I Legato*, par exemple —, il suffit de charger un *patch à articulations multiples* dans notre échantillonneur. Un *patch à articulations multiples* contient plusieurs articulations, et il est possible de passer de l'une à l'autre grâce aux *keyswitches*. Il s'agit *grosso modo* de plusieurs *patches* combinés en un seul pour plus de jouabilité. Pour reprendre notre exemple des *Violons I*, le *Do₀ — C0* en notation anglo-saxonne⁶⁵ — permet de basculer sur l'articulation *Legato*, tandis que le *C#0* est assigné à l'articulation *Staccato*, et ainsi de suite.

L'utilisation basique des *keyswitches* présente toutefois quelques inconvénients : par exemple, un *keyswitch* doit être joué avant la première note concernée par le changement d'articulation, ce qui, dans le cas de changement ou d'alternance de modes de jeu rapides, ou pour des compositeurs qui ne seraient pas très à l'aise au clavier, implique qu'il faille les ajouter dans un second temps, alloué spécifiquement à la « programmation MIDI » du morceau. Par ailleurs, si nous souhaitons transposer dans une autre tonalité une section musicale dans laquelle se trouvent des *keyswitches*, ces derniers seront aussi transposés, et la programmation MIDI, c'est-à-dire l'interprétation de cette section musicale, ne correspondra plus du tout à ce que nous souhaitions ; il faudra rééditer les *keyswitches* afin qu'ils retrouvent leur valeur initiale.

Pour une STAN, un *keyswitch* est une information de note MIDI comme une autre : elle ne fait donc pas la différence entre une note effectivement jouée par l'instrument virtuel et un *keyswitch*. Si le principe des *keyswitches* a été pensé pour que les *patches à articulations multiples* soient facilement jouables, leur gestion peut s'avérer laborieuse et particulièrement chronophage. Des solutions, qui s'appuient sur ce principe mais qui se veulent plus « pratiques », c'est-à-dire plus rapides et plus intuitives pour l'utilisateur,

⁶⁵ La notation anglo-saxonne est utilisée dans toutes les STAN pour désigner le nom des notes. La lettre *A* correspond au *La*, *B* au *Si*, *C* au *Do*, *D* au *Ré*, *E* au *Mi*, *F* au *Fa*, et *G* au *Sol*. Pour altérer une note, il suffit d'accoler l'altération (*#* ou *b*) à la lettre qui définit la hauteur de la note ; par exemple, la note *Fa dièse* est noté *F#*.

ont notamment été proposées par *Steinberg*, ou encore par *Babylonwaves*. En d'autres termes, ces deux sociétés ont développé des fonctionnalités avancées pour la gestion des articulations des instruments virtuels basées sur le système des *keyswitches*.

Très tôt dans son histoire, *Steinberg* a proposé des fonctionnalités à l'endroit de la gestion et de l'édition MIDI à la pointe de la technologie alors disponible. Dans *Cubase*, les articulations musicales, aussi appelées des « expressions », « définissent comment certaines notes doivent sonner ou être interprétées sur un instrument donné »⁶⁶. Ces expressions se subdivisent en deux types : les « directions » et les « attributs ». Les *directions* « sont valables pour toutes les notes à partir de leur position d'insertion, jusqu'à ce qu'une autre *direction* soit mentionnée dans la partition »⁶⁷ ; par exemple, « *legato* » peut être une *Direction*. Les *attributs*, quant à eux, « appartiennent à des notes séparées ; ils ne sont appliqués qu'aux notes pour lesquelles ils sont insérés »⁶⁸ ; il peut s'agir d'articulations comme *staccato*, *marcato*, *spiccato*, etc. L'utilisateur peut choisir de considérer une articulation donnée comme une *direction* ou comme un *attribut*.

Pour une meilleure gestion des articulations, *Cubase* a donc créé des « *Expression Maps* », ou « cartes d'expression ». Ces cartes définissent « le mappage [c'est-à-dire la cartographie] et les caractéristiques du son pour toutes [nos] expressions musicales. [...] Quand [nous sélectionnons] une *Expression Map* pour une piste MIDI ou une piste d'instrument, les articulations [c'est-à-dire les sons] définies dans la *map* sont automatiquement appliquées pendant la lecture. *Cubase* reconnaît les expressions inscrites pour le conteneur MIDI et recherche un son répondant aux critères définis dans [...] l'*Expression Map* »⁶⁹. Autrement dit, une *Expression Map* spécifique est nécessaire pour piloter chaque *patch* à articulations multiples. Une fois que

66 https://steinberg.help/cubase_pro_artist/v9/fr/cubase_nuendo/topics/expression_maps/expression_maps_arcticulations_about_c.html (dernière consultation le 29/04/2018).

67 *ibid.*

68 *ibid.*

cette *map* a été configurée, il est possible de passer d'une articulation à l'autre grâce à des lignes d'automatisation dédiées. Les *Expression Maps* permettent de gagner en visibilité, en clarté, en vitesse et en efficacité, mais aussi de nous affranchir d'éventuels problèmes de transposition des *keyswitches*, puisque ces derniers sont utilisés en sous-couche, et non plus comme des événements de notes à proprement parler.

Quant à *Babylonwaves*, qui propose d'ailleurs depuis quelques semaines des *Expression Maps* déjà configurées pour plus de quatre-vingt banques de son, ils ont surtout développé une alternative aux *keyswitches* pour *Logic Pro X : Art Conductor Logic*⁷⁰. Avant sa version 10.4, sortie au début de l'année 2018, *Logic Pro X* ne proposait pas de fonctionnalités facilitant la gestion des articulations. Avec l'ajout récent de ses « jeux d'articulations », semblables aux *Expression Maps* de *Cubase*, *Logic* a fait un grand pas en avant dans ce domaine ; toutefois, « il y a bien des choses à améliorer, même si la fonctionnalité [des jeux d'articulations] est réellement prometteuse et devrait améliorer le quotidien de plus d'un programmeur MIDI »⁷¹. Depuis cette version 10.4, *Art Conductor Logic* utilise le système des jeux d'articulations, mais auparavant, il fonctionnait grâce au *Scripter* de *Logic Pro X*. Le *Scripter* est un *plug-in* MIDI, c'est-à-dire qu'il peut être inséré sur une piste *Instrument* ou sur une piste *MIDI* pour traiter ou générer des données MIDI jouées par une région⁷² MIDI ou par le biais d'un clavier en temps réel. L'interface de programmation de *Scripter* est orientée « objet » et fonctionne avec *JavaScript*⁷³. Schématiquement, *Scripter* permet d'indiquer à quel *keyswitch* correspond chaque articulation. Dans les faits, on définit autant de variables pour le paramètre *Articulation* de notre instance de *Scripter* qu'il y a de *keyswitches* dans le *patch* qui doit être joué. Ce paramètre est ensuite

69 https://steinberg.help/cubase_pro_artist/v9/fr/cubase_nuendo/topics/expression_maps/expression_maps_expression_maps_c.html (dernière consultation le 29/04/2018).

70 <https://www.babylonwaves.com/logic-pro/> (dernière consultation le 29/04/2018).

71 *Test de Logic Pro X 10.4 d'Apple : Logic implacable* publié le 18/04/2018 sur *Audiofanzine* : <https://fr.audiofanzine.com/sequenceur-generaliste/apple/logic-pro-x/editorial/tests/logic-implacable.html> (dernière consultation le 30/04/2018).

automatisé, et l'utilisateur peut passer d'une articulation à l'autre en éditant sa courbe d'automatisation, qui a autant de pas que *Scripter* a de variables. A l'instar des *Expression Maps*, cette méthode présente un certain nombre d'avantages par rapport aux simples *keyswitches*, notamment en termes d'efficacité et de visibilité.

Bien évidemment, les *keyswitches* ne constituent pas l'unique alternative pour la gestion des articulations et des modes de jeu d'un instrument virtuel. L'autre possibilité consiste tout simplement à utiliser les *patches* qui n'ont qu'une seule articulation, et de créer autant de pistes *MIDI* que nous avons besoin d'articulations pour chaque instrument. Dans un morceau où le premier

72 On parle de « région audio » ou de « région MIDI » dans *Logic Pro* comme on parle de « clip audio » ou de « clip MIDI » dans *ProTools* pour désigner « un segment rectangulaire affiché dans la zone de Pistes — communément appelée la « fenêtre d'arrangement » — représentant les enregistrements et les fichiers multimédias d'un projet, et servant de conteneur pour les données audio ou MIDI. [...] une région MIDI est donc un « conteneur de données pour les événements MIDI, affiché sous forme de zone rectangulaire arrondie dans la zone de Pistes. » — D'après le *Guide de l'utilisateur de Logic Pro X* (version 10.3).

73 *JavaScript* est un langage de programmation de scripts orienté objet à prototype, ce qui signifie que les bases du langage et ses principales interfaces sont fournies par des objets équipés de constructeurs qui permettent de créer leurs propriétés. L'une des principales caractéristiques des objets est le « prototypage », qui permet de créer des objets personnalisés. Historiquement, un *script* permettait de lancer et de coordonner l'exécution d'un programme. Aujourd'hui, par abus de langage, on désigne par « langage de script » n'importe quel langage de programmation qui est *interprété*, en opposition aux langages qui sont *compilés*, comme le *C++* ou le *Java* — à ne pas confondre avec le *JavaScript* ! Pour schématiser, lorsqu'un programme est exécuté par le biais d'un interpréteur, les opérations d'analyse et de traductions nécessaires à l'exécution du programme par une machine sont réalisées à *chacune* de ses exécutions ; *a contrario*, le compilateur traduit une seule et unique fois le code source du programme écrit dans un langage de haut niveau vers un langage de bas niveau, aussi appelé « langage machine ». Une fois que le programme a été compilé, il peut être exécuté immédiatement par une machine, ce qui n'est pas le cas d'un langage comme *JavaScript*, qui a systématiquement besoin d'un interpréteur. Pour finir, *JavaScript* permet au développeur de s'affranchir des contraintes de bas niveau, qui sont prises en charge par l'interface, tout en bénéficiant d'une syntaxe de haut niveau.

violon jouerait des parties *legato*, *staccato* et *pizzicato*, nous aurions besoin de trois pistes MIDI distinctes. Il est possible de créer trois pistes *Instrument* ou trois pistes *MIDI* pilotant un instrument multi-timbral ; le résultat sera le même, et ce choix n'aura d'influence que sur notre configuration et la gestion de la mémoire vive de notre ordinateur. Un instrument virtuel « multi-timbral » désigne un lecteur d'échantillons, comme *Kontakt* ou *Play*, dans lequel se trouvent plusieurs *patches* et qui peut avoir plusieurs sorties audio. Une entrée MIDI est définie par un *port* et par un *canal* ; un port MIDI peut contenir jusqu'à seize canaux.

Reprenons l'exemple des premiers violons, et assignons l'échantillonneur multi-timbral au *port 1* de notre studio MIDI : nous pourrions fixer l'entrée du premier patch (*legato*) sur le *canal 1*, le second (*staccato*) sur le *canal 2*, et le dernier (*pizzicato*) sur le 3. Dans ce cas, la sortie de la piste MIDI qui doit nous permettre de jouer le patch « *Violons I Legato* » est assignée au *canal 1* du *port 1* de notre STAN, la piste « *Violons I Staccato* » au *canal 2*, et la piste « *Violons I Pizzicato* » au *canal 3*. Enfin, les différentes sections musicales du morceau seront découpées et placées sur les pistes correspondantes à l'articulation souhaitée. Ainsi, nous aurons par exemple les mesures 1 à 4 sur la première *piste MIDI* (*legato*), 5 à 8 sur la troisième (*pizzicato*), la neuvième sur la seconde (*staccato*), et la fin du morceau sur la première (*legato*).

L'utilisation des *keyswitches* ou de *patches* à articulation unique se valent en termes de résultat. Chaque compositeur a sa préférence, et il est même tout à fait possible de les combiner en fonction de chaque instrument, de la manière de programmer de tel ou tel éditeur, et bien entendu des habitudes de travail de chacun.

3.2. Gestion des nuances et de la dynamique

Si la gestion des articulations et des modes de jeu est cruciale, tant sur le plan musical qu'acoustique, celui de la dynamique est également essentiel à la réussite d'une maquette MIDI. La dynamique englobe à la fois l'écart entre le plus petit et le plus haut niveau sonore d'un morceau, mais aussi la *nuance*, c'est-à-dire le volume auquel joue un instrument. Cette nuance a un effet plus ou moins marqué et marquant sur le timbre d'un instrument : un trombone qui joue *fortissimo* n'est pas seulement beaucoup plus fort qu'un trombone qui joue *pianissimo*, mais il a aussi un timbre relativement différent. Là encore, il existe deux méthodes pour gérer la dynamique d'un instrument virtuel, qui est en général fonction de l'articulation qui est employée : (i) la *vélocité*, et (ii) la *modulation*, qui est assignée au *Control Change 1 (CC#1)*.

En MIDI, un *control change*, abrégé *CC*, est un contrôle continu : il permet de piloter le paramètre de jeu auquel il est assigné avec une résolution de 0 à 127. La norme MIDI prévoit cent vingt-huit *CC*, dont certains sont assignés par défaut à un paramètre de jeu, comme la *modulation (CC#1)*, le *volume (CC#7)*, l'*expression (CC#11)*, le *panoramique (CC#10)*, ou encore la pédale de maintien, ou de « *sustain* » (*CC#64*), etc. Un grand nombre des cent vingt-huit *Control Changes* sont vides, et peuvent être assignés à n'importe quel paramètre que l'utilisateur souhaite piloter avec le contrôleur MIDI de son choix.

Nous ne pouvons pas entrer dans les détails de fabrication d'un instrument virtuel, mais pour schématiser, une même note est composée de plusieurs échantillons. Ces échantillons permettent notamment de gérer le placement de l'instrument dans un espace acoustique — si la banque propose plusieurs positions de microphones —, mais ils permettent surtout de jouer sur la nuance de l'instrument. Plus l'instrument joue fort, plus son volume augmente, mais surtout, des échantillons enregistrés dans la nuance musicale correspondante sont déclenchés. Autrement dit, ce ne sont pas les mêmes échantillons qui sont déclenchés lorsque l'instrument joue *piano*, *mezzo forte*

ou encore *fortissimo* ; il y a donc plusieurs « couches » d'échantillons pour une même note, et ces couches sont parcourues et combinées en fonction de la nuance souhaitée. Tout le problème réside donc dans la fluidité et la musicalité du parcours de ces couches et du passage de l'une à l'autre, afin d'imiter au mieux les changements de jeu et de nuance d'un instrumentiste, que ce soit sur la longueur d'un morceau, ou sur une seule phrase musicale. L'objectif est bien entendu que le passage d'une couche à l'autre soit particulièrement fluide et naturel, autrement dit qu'il soit parfaitement transparent pour l'oreille humaine.

La *vélocité* est en général utilisée pour les modes de jeu courts, comme *spiccato* ou *staccato*, puisqu'elle est encapsulée dans le message de *Note In*, ce qui induit qu'elle est fixée au moment de l'émission de la note, et qu'elle ne peut donc pas évoluer dans le temps, à moins que la note ne soit jouée à nouveau. La *vélocité* est comprise entre 0 et 127, et plus elle est élevée, plus la nuance de la note est importante. *A contrario*, la *modulation (CC#1)* est préférée pour les modes de jeu long, comme *legato* ou *sostenuto*, puisqu'en tant que contrôle continu, sa valeur peut évoluer alors que la note est tenue, le message de *Note In* et celui de *Control Change* étant complètement dissociés. Autrement dit, le *CC#1* permet de faire évoluer la nuance et le timbre d'une note dans le temps, et ainsi construire des *crescendi* ou des *decrescendi*. Il est possible de contrôler des modes de jeu courts avec le *CC#1* et des modes de jeu longs avec la *vélocité*, mais nous comprenons bien que pour des raisons musicales évidentes, cela ne fait pas forcément sens. Le *CC#1* pourrait contrôler l'ensemble des modes de jeu, mais la *vélocité* est tout de même plus pratique pour les modes de jeu courts. En fin de compte, nous pourrions faire l'analogie entre les *Expression Maps* de *Cubase* et le choix de la *vélocité* ou de la *modulation* pour la gestion de la dynamique et des nuances : la *vélocité* se comporte comme un *attribut*, puisqu'elle n'a d'influence que sur la note à laquelle elle est assignée, tandis que la *modulation* fonctionne comme une *direction*, ce qui signifie que sa valeur s'applique à toutes les notes jouées par l'instrument qu'elle pilote.

3.3. Pré-mixage et finalisation

Une STAN gère à la fois les données audio et MIDI, et avec ou sans *Vienna Ensemble Pro*, le compositeur peut intervenir à tous les niveaux du chemin du signal et à toutes les étapes de composition et de production d'une *cue* : de la programmation MIDI à la sortie audio de chaque instrument virtuel, des chaque *stem* au *master* — le *master* correspond au canal qui fait la sommation stéréophonique ou multi-canal du mixage final de la maquette. En ce qui concerne le traitement des données audio, le compositeur a accès à tous les outils disponibles dans une STAN et utilisés par n'importe quel mixeur : modules d'égalisation, de compression, de réverbération, de délai, etc. Il est donc en mesure de prémixer sa maquette comme bon lui semble et en fonction de ses compétences. Nous ne nous attarderons donc pas sur le processus de (pré)mixage en tant que tel, mais avant de clore ce chapitre, nous voudrions préciser quelques éléments sur la programmation MIDI et sur les problématiques que peut engendrer la production d'une maquette dans la relation entre le réalisateur et le compositeur.

La programmation MIDI consiste à éditer et automatiser tous les paramètres MIDI qui nous permettent de jouer sur la musicalité, l'interprétation, la vraisemblance, ou encore la dynamique de notre maquette. Le choix des articulations, des nuances, grâce à la vitesse ou à la modulation, font partie intégrante de ce processus. La programmation permet aussi de faire du prémixage sans pour autant exploiter les fonctionnalités liées au traitement de l'audio. En effet, la plupart des compositeurs ne mixent pas la piste audio correspondant à la sortie d'un instrument virtuel, mais préfèrent moduler le niveau de sortie de l'échantillonneur. Cela est possible grâce à deux *Control Changes* : le *volume* (CC#7) et l'*expression* (CC#11).

Dans son ouvrage, Paul Gilreath préconise d'utiliser le *Volume* afin de fixer le niveau moyen d'un instrument virtuel. Une fois que la valeur du CC#7 a été choisie pour un instrument donné, il recommande de ne plus y toucher, et d'utiliser « seulement le CC#11 pour contrôler le volume », c'est-à-dire pour

ajouter des *crescendi* et des *decrescendi*. Il ajoute qu'il est préférable qu'une phrase musicale débute au milieu de la dynamique possible de l'instrument, ce qui implique que le *CC#11* soit au milieu de sa course ($CC\#11 \approx 64$), afin de pouvoir aisément moduler autour de cette valeur moyenne⁷⁴. En réalité, l'*Expression* correspond à un pourcentage du *Volume* ; le *CC#11* est donc plus précis que le *CC#7*, ou du moins permet de fluctuer autour de sa valeur, tout en laissant le niveau de sortie de l'échantillonneur — piloté par *CC#7* — intact.

Le protocole MIDI mémorise la valeur d'un paramètre jusqu'à ce que celle-ci change. Un changement de valeur n'est pris en compte que lorsqu'il est « joué », c'est-à-dire au cours de la lecture. Bien que les *Expression Maps* aient permis de s'affranchir de ce problème pour les articulations, il n'en demeure pas moins que le programmeur MIDI doit garder cette problématique à l'esprit dans bien des cas, notamment pour la gestion des contrôles continus.

Pour illustrer cela, prenons l'exemple des *keyswitches*. Nous avons expliqué que dans son fonctionnement basique, un *keyswitch* doit être placé avant la première note à laquelle doit être appliquée la nouvelle articulation. Admettons que nous ayons quatre mesures avec un changement d'articulation au début de la troisième. Un premier *keyswitch* est placé avant la *mesure 1*, et un second entre la *mesure 2* et la *mesure 3*. Si nous lisons une première fois ces quatre mesures depuis le début, les deux *keyswitches* vont s'appliquer correctement. Si par contre dans un second temps nous relançons la lecture depuis le début de la *mesure 2*, elle sera jouée avec la seconde articulation, puisque le *keyswitch* de la *mesure 1*, censé être appliqué à ce moment-là, n'aura pas été joué.

74 « *CC#7* is the continuous controller for *Channel Volume* and is used to set the channel's main volume. *CC#11* is the continuous controller for *Expression*, and is typically used to incorporate crescendo and decrescendo swells into the phrase. Once *CC#7* is set, leave it alone: you will then use only *CC#11* for controlling volume. [...] Get into the habit of starting phrases in the middle of your sample's dynamic/layer range. This gives you room to get louder or softer as the phrases require. » — Paul Gilreath, *op. cit.*, pp. 238-239.

Pour conclure, le pré-mixage et la finalisation d'une maquette MIDI sont deux étapes essentielles, qui peuvent être déterminantes dans la validation d'un *cue* par la réalisation et la production. Richard Davis compare une maquette à une épée à double tranchant : « d'un côté, elle permet de donner au réalisateur une idée assez précise de la direction prise par le compositeur pour le *cue*, et au compositeur de s'assurer que sa proposition est conforme à la vision du réalisateur. Le réalisateur peut faire des suggestions et offrir des commentaires, se sentir inclus dans le processus de composition, et ainsi permettre au compositeur et à ses collaborateurs d'être rassurés quant au fait que tout le monde se dirige dans la bonne direction. D'un autre côté, il peut être très inconfortable d'avoir le réalisateur entrain de regarder littéralement par-dessus notre épaule en faisant des suggestions musicales. De plus, dans cette version sommaire du *cue*, le réalisateur peut se focaliser uniquement sur le son synthétique des cordes ou le timbre peu réaliste des cors, et penser que la musique ne fonctionne pas, sans pour autant avoir la compétence musicale nécessaire pour se projeter et imaginer ce que donnera le *cue* enregistré par de véritables musiciens. [...] Les attentes du monde de l'industrie cinématographique ne concernent pas seulement la nécessité de composer la musique d'un film [très] rapidement, mais aussi qu'elle sonne bien sous la forme d'une maquette dans le studio du compositeur. »⁷⁵. Il ne faut donc en aucun cas négliger la production d'une maquette, c'est-à-dire son (pré)mixage et sa couleur sonore ; dans bien des cas, elle est déterminante pour la

75 « On the one hand, it can give the director a good idea of where the composer is going with the cue, and the composer can be assured that the director's vision is being accomplished. The director can offer suggestions and comments, feel involved in the music process, and leave the composer's studio feeling secure that the music is going to the right direction. On the other hand, it can be very uncomfortable to have the director literally standing over one's shoulder making musical suggestions. In addition, on this rough version of the cue, the director might hear only the electronic sounding synth strings and not-quite-real sounding French horn sample and think is it terrible, not having the musical ability to make the imaginary transfer to real instruments. [...] The expectation of today's film-making world is not only that the music will be composed very quickly, but that it will sound great as a mock-up in the composer's studio. » — Richard Davis, *op. cit.*, pp. 96-97, et 99.

validation d'un *cue*. Pour aller encore plus loin, il n'est pas rare qu'aujourd'hui, la production et le réalisateur s'attendent à ce que des maquettes sonnent comme la musique définitive qui sera dans la bande son du film.

Dans certains cas, le rejet d'un *cue* ne relève pas de son contenu musical, mais uniquement de son rendu sonore. D'un autre côté, une maquette « trop réussie » pourra remplacer la version enregistrée d'un *cue* dans la version finale de la bande son du film, car le réalisateur l'aura davantage dans l'oreille que la version enregistrée par de vrais musiciens. Autrement dit, le maquettage d'un *cue* a des avantages et des inconvénients, et si le compositeur doit y prêter assez d'attention, il ne doit pas pour autant en perdre par excès de zèle. En général, ce sont ses assistants qui se chargent de la finalisation de la programmation MIDI, tandis qu'il travaille sur un autre *cue*. Enfin, il peut être nécessaire de prémixer la maquette d'un *cue* comme le ferait le mixeur avec le montage du son du film, afin que le réalisateur puisse plus aisément se projeter et se rendre compte de l'équilibre établi entre la musique et le reste de la bande son du film.

C. Avantages et inconvénients des outils modernes

Nous avons tenté de décrire assez précisément les méthodes de travail du compositeur de musique de film moderne, ainsi que les outils qu'il convoque dans son processus de création et de production. Si les méthodes de composition et de production actuelles présentent un certain nombre d'avantages par rapport aux méthodes plus « traditionnelles », elles ont aussi des inconvénients et induisent de nouvelles problématiques. Parmi elles se trouvent les spécificités de l'orchestration MIDI par rapport à l'orchestration pour de véritables instrumentistes, ainsi que la question de l'humanisation des instruments virtuels. Enfin, nous verrons que la quantification est au cœur de cette question de l'humanisation, et qu'elle constitue un paramètre dont la gestion est globalement peu automatisable. De ce fait, la quantification est encore aujourd'hui embarrassante et freine le passage d'un fichier maqueté à la phase d'orchestration, c'est-à-dire le transfert de la STAN à la partition.

1. Similarités et spécificités de l'orchestration MIDI

Lorsqu'il maquette, le compositeur préorchestre chaque *cue* dans la mesure du possible. Sa *template* et ses contrôleurs MIDI lui permettent d'enregistrer rapidement de nombreuses et diverses parties instrumentales, afin d'obtenir une maquette qui tende autant que possible vers la version finale du *cue* qui sera enregistré par un véritable ensemble instrumental. Cependant, il joue toutes les parties instrumentales grâce à son clavier maître, ce qui signifie qu'il peut aussi bien « jouer » du piano que de la harpe, du hautbois ou du violoncelle. Cela implique qu'il peut jouer des éléments musicaux qui sont physiquement ou organologiquement impossibles à réaliser sur un véritable instrument. Par exemple, un *glissando* chromatique à la harpe, ou certaines double-cordes au violon, ou plus simplement des notes tenues beaucoup trop longues ou des traits virtuoses interminables sans prévoir de respiration pour

un instrument à vent, etc. D'une part, tous ces éléments participent au réalisme et à la vraisemblance d'une maquette : notre oreille est habituée à entendre des *glissandi* de harpe, mais elle n'a jamais pu entendre de *glissandi* chromatiques joués par une seule harpe ; inconsciemment, il sonnera donc « faux », même si notre harpe virtuelle est bien échantillonnée. D'autre part, il semble évident qu'écrire des éléments qui ne sont pas jouables pour de véritables musiciens est un problème en soi. Dans ce cas, il revient à l'orchestrateur de proposer des solutions alternatives pour obtenir un effet similaire, mais jouable.

Cela étant, certains compositeurs se servent de cette souplesse des instruments virtuels afin d'obtenir un effet particulier : c'est le cas de John Williams, qui, pour la musique qu'il a composée pour la saga *Harry Potter*⁷⁶, a combiné un célesta virtuel et un véritable célesta afin d'obtenir un effet plus aérien et encore plus « magique » que celui qui est déjà habituellement attribué au célesta. Aujourd'hui, les instruments de l'orchestre sont très bien échantillonnés, mais pour qu'une maquette soit vraisemblable, il est nécessaire que le compositeur respecte les contraintes organologiques spécifiques à chaque instrument. Il mobilise donc toutes ses connaissances d'instrumentation, d'orchestration et d'organologie dès la composition, et ne doit surtout pas se laisser prendre au jeu du « tout-jouable » induit par l'emploi d'instruments virtuels, sauf à dessein.

76 John Williams a composé la bande originale des trois premiers volets de la saga : *Harry Potter and the Philosopher's Stone* (« *Harry Potter à l'école des sorciers* », Chris Columbus, 2001), *Harry Potter and the Chamber of Secrets* (« *Harry Potter et la Chambre des secrets* », Chris Columbus, 2002) et *Harry Potter and the Prisoner of Azkaban* (« *Harry Potter et le Prisonnier d'Azkaban* », Alfonso Cuarón, 2004).

Tristan Noon pointe sans doute ce qui est le plus symptomatique du « tout-jouable » ou du « pré-orchestré » introduit par les instruments virtuels : les « redoutés accords du *patch* d'ensemble »⁷⁷. Il s'agit d'un accord joué grâce à un *patch* d'ensemble sur une seule piste — par exemple, le quintette à cordes au complet, voire même l'orchestre dans son intégralité — comme s'il s'agissait d'une partie de piano ; dans ce cas, l'orchestrateur doit répartir toutes les notes de l'accord entre toutes les voix de l'ensemble instrumental qu'il juge adéquat. Les *patches* d'ensemble regorgent aussi d'effets d'orchestre préenregistrés : il suffit au compositeur de presser une ou seulement quelques touches de son clavier maître pour déclencher un effet d'orchestre complet et complexe. Ces effets sont très efficaces et très utilisés dans la musique de film, mais il serait trop compliqué de les reproduire avec des instruments virtuels individuellement : c'est pour cette raison que les éditeurs de banques de son les préenregistrent. Ils sont donc très facilement utilisables par le compositeur, mais il est tout de même préférable de les réenregistrer, tout d'abord parce que sinon tous les compositeurs auraient *exactement* les mêmes effets dans leurs bandes originales, et ensuite parce qu'un effet préenregistré ne correspond pas forcément parfaitement à ce que souhaite le compositeur. L'orchestrateur doit donc relever à l'oreille toutes les voix nécessaires à la réalisation de l'effet, et le reproduire, ou en proposer une variation.

Pour résumer, les instruments virtuels présentent l'avantage de pouvoir produire des maquettes vraisemblables, qui permettent au réalisateur de valider la musique qui accompagnera une scène de son film en toute connaissance de cause, et ainsi éviter de mauvaises surprises au cours de la séance d'enregistrement. De plus, les éditeurs d'instruments virtuels proposent de nombreux outils de préorchestration, qui offrent un gain de temps considérable au compositeur, qui de son côté ne doit pas tomber dans l'écueil

77 « As an orchestrator, you have to be prepared to face what I call the “dreaded ensemble patch chords”. This is where the composer [...] has just played in an enormous piano-like chord with two hands to save time. It is then up to you as the orchestrator to split this out into something playable. » — Tristan Noon, *op. cit.*, p. 68.

du « tout-jouable », et conserver autant que possible une approche organologique des instruments virtuels qui sont à sa disposition.

2. L'humanisation, ou la recherche de la vraisemblance musicale

Nous l'aurons compris, le but d'une maquette MIDI est de tendre vers le réalisme ou la vraisemblance. Pour cette raison, la question de l'humanisation est au cœur de la programmation MIDI. Il existe donc un certain nombre de techniques et diverses astuces de programmation pour contourner les faiblesses des instruments virtuels. Nous ne pourrions pas détailler ici toutes les techniques de programmation pour tous les types d'instruments, et nous nous contenterons d'aborder la programmation du *legato*, de l'effet « mitraillette » que peuvent introduire les articulations courtes, et enfin de l'automatisation de *tempo*.

Dans une STAN orientée vers la composition musicale, l'alignement des évènements MIDI, notamment des notes, se fait sur une grille musicale. Cette grille est fonction de la signature rythmique, ainsi que d'une résolution temporelle, qui correspond à un pas de quantification donné en PPQN. Chaque évènement est alors adressé grâce à un numéro de mesure, un temps musical, une division, et éventuellement un nombre de *ticks*. Les données de transport et de position des évènements dans une STAN, quels qu'ils soient, ont donc la forme suivante : « *bar.beat.div.tick* », ou « *measure.temps.div.tick* » ; cet adressage temporel n'est pas sans rappeler la nomenclature du TC.

Comme le compositeur joue au clavier chaque partie, les notes ne sont pas exactement alignées sur la grille musicale. Cependant, de vrais musiciens ne sont pas non plus absolument alignés sur cette grille, car ce sont justement ces légères fluctuations autour d'une grille absolue et parfaitement régulière qui donnent la « vie » et la musicalité d'un morceau ; ces fluctuations sont le privilège de l'interprétation. Autrement dit, dans un souci de vraisemblance et

de musicalité, il ne faut pas aligner systématiquement les notes sur la grille musicale, et conserver ces [très] légers décalages. En général, on les conserve pour les modes de jeux longs, et on préfère tout de même aligner les articulations courtes, afin de construire une assise rythmique solide.

En ce qui concerne la programmation d'un mode de jeu *legato*, il est généralement nécessaire de débiter la note quelques *ticks* avant sa position de départ absolue, afin de compenser son attaque qui peut manquer de franchise. Il est aussi préférable de laisser deux notes successives se superposer légèrement, afin que la transition de l'une à l'autre soit la plus naturelle possible. Typiquement, « dix à quinze *ticks* de chevauchement sont suffisants pour permettre une bonne transition d'une note à l'autre »⁷⁸. Certaines banques de cordes frottées offrent même la possibilité d'ajouter le glissement du doigt de l'instrumentiste sur la corde pour passer d'une note à l'autre en fonction de ce chevauchement et de la vitesse de chaque note. En règle générale, les articulations longues, et particulièrement le *legato*, sont plus difficiles à programmer que les articulations courtes ; d'une part leur émulation est plus compliquée, et d'autre part, elles induisent des intentions et des gestes musicaux plus complexes. Un certain nombre de contrôleurs MIDI, comme la molette de modulation, ou encore un contrôleur de respiration, peuvent être utilisés pour les piloter et les programmer avec plus d'efficacité, de musicalité et de réalisme.

Si les articulations courtes, comme *staccato* ou *spiccato*, sont plus faciles à émuler, leur programmation peut elle aussi présenter quelques difficultés. La plus proéminente est sans doute celle que l'on appelle « l'effet mitraillette » ou « l'effet mitrailleuse ». Cet effet est produit par des notes répétées rapidement⁷⁹ : il est d'autant plus prégnant lorsqu'il s'agit d'une même note, mais il apparaît aussi pour des notes différentes répétées alternativement. Il

78 « Typically, ten to fifteen ticks of overlap are sufficient to allow for a nice transition from note to note. » — Paul Gilreath, *op. cit.*, p. 235.

79 « In the sampling world, repeated notes can exhibit what is commonly known as the *machine gun effect*. » — Paul Gilreath, *op. cit.*, p. 232.

est plus précisément dû à la répétition d'un même échantillon. Par exemple, si une trompette ne présente qu'un seul échantillon pour son articulation *staccato forte*, et qu'elle joue en croches la même note, par exemple un *Ré*, alors chaque *Ré* sera *exactement* le même. L'oreille humaine est sensible à la répétition d'un même son, et identifie cet effet de boucle ou de clonage rapidement, pour ne pas dire immédiatement. Si cette répétition se déroule à un *tempo* relativement rapide, alors elle pourra avoir l'effet d'une rafale de mitrailleuse, d'où ce sobriquet.

Les éditeurs et les programmeurs MIDI ont trouvé plusieurs solutions pour s'affranchir de ce problème : la première consiste à changer la vitesse d'une note à l'autre, afin de ne pas utiliser la même couche d'échantillons, mais cela n'est pas toujours possible musicalement. La seconde, qui est la plus répandue et la plus utilisée, consiste à activer le *Round Robin*. L'option *Round Robin* propose plusieurs échantillons pour une même articulation et une même note, et permet d'alterner aléatoirement entre ces échantillons. Les banques de son de bonne facture le proposent systématiquement pour les articulations courtes, et leur qualité dépend du nombre d'échantillons proposés par articulation et par nuance. Enfin, il est possible de simuler le *Round Robin* en utilisant les échantillons voisins de la note que nous voulons répéter, et de les transposer afin d'avoir plusieurs échantillons. Par exemple, pour un *Ré*, nous pourrions simuler un *Round Robin* en transposant le *Do dièse* au demi-ton supérieur et le *Mi bémol* au demi-ton inférieur, afin d'obtenir trois échantillons différents pour le *Ré*. Cette méthode est moins élégante, mais elle a le mérite de fonctionner dans de nombreux cas ; il faudra par exemple éviter de transposer des notes jouées sur la corde à vide d'un violon, dont la sonorité diffère des positions intermédiaires. Notons tout de même que cette méthode de transposition des échantillons voisins était surtout utilisée il y a quelques années, lorsque peu d'éditeurs proposaient un *Round Robin* pour les articulations courtes.

Enfin, la clé de l'humanisation d'une maquette MIDI réside sans doute dans l'automatisation du *tempo*. Comme le souligne Paul Gilreath, « le tempo d'une musique jouée par un orchestre n'est jamais statique »⁸⁰. À l'instar de la première piste d'un fichier MIDI de *Format 1*, qui contient tous les événements méta globaux, notamment les variations de signature rythmique et de *tempo*, une STAN dispose de pistes globales. Parmi ces pistes globales, se trouvent évidemment une piste de signature rythmique, et une piste de *tempo* automatisables. Nous pouvons distinguer deux usages principaux de la variation du *tempo* dans le cadre de la musique à l'image : (i) des variations musicales, à la manière des *ritardandi* ou des *rubati* que nous pouvons trouver dans la musique dite « classique », afin de mettre en valeur le discours musical, et (ii) des variations narratives, dans le but de créer quelques points de synchronisation entre la musique et l'image. Bien évidemment, les variations de *tempo* dites « musicales » ne doivent pas se faire au détriment de la synchronisation de la musique et de l'image, c'est-à-dire au détriment de la narration et de la dramaturgie du film.

80 « The tempo of music played by a live orchestra is never static. » — Paul Gilreath, *op. cit.*, p. 373.

Quoiqu'il en soit, toutes les variations de *tempo* se divisent elles aussi en deux catégories : les variations abruptes et les variations graduelles. Les premières se produisent lorsque deux *tempi* différents sont accolés l'un à l'autre, tandis que les variations graduelles se produisent « typiquement dans de courtes phrases ou durant quelques mesures, et impliquent souvent un changement de la valeur du *tempo* initial, auquel on revient lorsque la variation est terminée »⁸¹. Dans le but de simuler les légères fluctuations de *tempo* introduites par un chef d'orchestre dirigeant un véritable orchestre, le programmeur MIDI peut insérer de très légers changements de *tempo* tout au long d'un morceau supposé être à *tempo* constant. En effet, « aucun chef d'orchestre et aucun orchestre ne peut jouer à un tempo constant sur une longue période (sauf s'ils jouent avec une piste de métronome), et de ce fait les *tempi* d'une performance orchestrale varieront quelque peu tout au long du morceau »⁸².

3. La quantification

Nous avons abordé la question de l'alignement ou non des événements MIDI sur la grille musicale. En MIDI, la *quantification* correspond à l'action d'aligner des événements sur la grille musicale, c'est-à-dire à effacer les imperfections rythmiques d'une performance jouée, ce qui revient finalement à la déshumaniser. Dans les faits, quantifier consiste à aligner tous les événements MIDI sur la grille musicale, ou du moins sur la division [musicale] appropriée. Si certains instruments sont quantifiés dès la composition, la

81 « All tempo changes fall into one of two categories: abrupt or gradual. Abrupt tempo changes occur when two different *tempi* are butted up against one another. Gradual changes typically occur in short phrases or measures and often involve making a change to the initial tempo and then returning to it after the change is completed. » — Paul Gilreath, *op. cit.*, pp. 374-375.

82 « Because no conductor and therefore no orchestra can play at a constant tempo for any period of time (unless playing to a click track), the *tempi* of orchestral performances will vary slightly throughout a piece. » — Paul Gilreath, *op. cit.*, p. 384.

plupart d'entre-eux sont, comme nous l'avons vu, fluctuants, c'est-à-dire non-quantifiés.

Nous avons expliqué précédemment que la phase d'orchestration d'un *cue* commence par l'import d'un fichier MIDI dans un logiciel de notation musicale, qui ne fonctionne qu'avec une grille musicale absolue — dans la suite, nous parlerons uniquement de *Sibelius*, mais son fonctionnement étant dans son principe similaire à ses concurrents, nous pouvons l'assimiler à un logiciel de notation musicale « universel » ; autrement dit, *Sibelius* est à entendre comme « logiciel de notation musicale » dans le contexte de ce mémoire. À l'import, *Sibelius* va transcrire toutes les fluctuations autour de la grille musicale comme si elles *devaient* être jouées telles quelles, ce qui est musicalement aberrant. Si une ronde déborde d'une dizaine de *ticks* du début de la mesure, cela ne signifie évidemment pas qu'elle doit être jouée en anacrouse. En d'autres termes, *Sibelius* interprète les fluctuations autour de la grille comme des décalages rythmiques qui doivent être notés, ce qui est absurde. Il est donc nécessaire de quantifier les événements MIDI avant d'importer le SMF dans *Sibelius*, afin que la notation qui en résulte soit cohérente musicalement.

Pour résumer, la maquette produite par le compositeur n'est pas, ou peu quantifiée pour des questions de vraisemblance musicale ; puis, elle est quantifiée pour que sa notation soit juste et lisible, et enfin, la musique est jouée et enregistrée par de véritables musiciens, qui réintroduisent la musicalité et les légères fluctuations de *tempo* qui avaient été perdues au cours de la quantification.

Il semble ici assez clair que le processus du passage de la STAN à la partition peut être relativement chronophage, et parfois même ardu. Bien que les STAN proposent toutes des outils pour quantifier les données MIDI, cette étape de quantification ne peut pas être complètement automatisée sur la longueur d'une session. Trop de paramètres sont impliqués et chaque instrument induit un trop grand nombre de cas particuliers pour qu'ils soient

tous généralisés au niveau macroscopique, c'est-à-dire au niveau de la session dans sa globalité. Aujourd'hui, il revient donc à l'assistant du compositeur ou à l'orchestrateur de quantifier les données MIDI de chaque *cue* avant de les importer dans *Sibelius*, et de commencer la phase d'orchestration.

Tristan Noon décrit le processus qu'il juge le plus rapide pour accomplir cette étape de quantification. Dans un premier temps, il faut quantifier le début de chaque note, afin qu'il soit aligné sur le bon temps musical. Une fois que le début de la note a été quantifié, il faut aussi en quantifier la fin, afin qu'elle s'achève aussi sur le bon temps musical. Généralement, cette quantification semi-automatique, réalisée par sections musicales, crée de légers chevauchements : il faut donc les supprimer — notons qu'en ayant quantifié le début et la fin de chaque note, les chevauchements de « *legato* » que nous avons décrits précédemment ont déjà été supprimés. Certains outils ou certaines fonctions, comme « forcer le *legato* » sont des alliés précieux, et permettent de gagner du temps. L'éditeur de partition de notre STAN pourra nous permettre de vérifier que les données quantifiées sont correctes avant d'être importées dans *Sibelius*⁸³.

Une fois que les données MIDI ont été correctement quantifiées, elles peuvent être importées sans encombre dans *Sibelius*, et le *cue* peut poursuivre son voyage vers la session d'enregistrement et le mixage final du film. Cependant, nous pensons qu'il pourrait être judicieux d'optimiser ce passage de la STAN à *Sibelius*, afin de dégager plus de temps de travail effectif à l'orchestrateur. Cette étape technique de quantification des données ne cristallise finalement qu'une perte de temps, à l'heure où les temps de production tendent à se réduire encore davantage. Une analyse moins globale d'un SMF, qui s'effectuerait au niveau de chaque instrument et par section pourrait nous permettre de quantifier automatiquement un SMF. Autrement dit, nous allons élaborer d'une interface logicielle entre la STAN et la partition, qui permette notamment une quantification adaptative des données MIDI.

83 Tristan Noon, *op. cit.*, pp. 42-54.

III. Développement d'un outil logiciel pour optimiser le passage de la STAN à la partition

A. Transfert de données d'une STAN vers *Sibelius* : les bases d'une interface plus adaptée

1. La liste d'évènements ou « *Event List* »

Pour rappel, les évènements MIDI contenus dans un SMF sont temporellement définis les uns par rapport aux autres : chaque évènement est composé de deux valeurs, à savoir un Δt et un bloc de données. Le Δt désigne le temps qui sépare l'évènement de celui qui le précède ; il est toujours donné en *ticks*, dont la valeur dépend de la division temporelle du SMF, renseignée dans son en-tête. Un *tick* est la plus petite valeur unitaire relative à cette division temporelle ; dans notre cas, c'est-à-dire dans le cas d'une division temporelle musicale, 1 *tick* = 1 PPQN. Dans les faits, la résolution d'un SMF est habituellement égale à 960 PPQN : dans ce cas, si deux évènements sont séparés par $\Delta t = 480$ PPQN, alors ils sont distants d'une croche. Afin de pouvoir déplacer et éditer aisément les données MIDI contenues dans un SMF, il est nécessaire de nous affranchir de la dépendance temporelle qui existe entre deux évènements consécutifs, c'est-à-dire passer d'une référence temporelle relative à une référence temporelle absolue. Cela revient à recalculer le Δt de chaque évènement MIDI non plus par rapport à l'évènement qui le précède, mais par rapport à l'origine temporelle du SMF, c'est-à-dire par rapport à *zéro*. Ainsi, le nouveau Δt absolu de chaque évènement MIDI, noté Δt_{abs} , correspond à l'écart de temps qui le sépare du début du fichier.

Dans le cadre du prototypage de notre interface logicielle entre la STAN et la partition — autrement dit entre la STAN et *Sibelius* —, nous allons uniquement nous préoccuper des évènements de notes, c'est-à-dire en somme des messages *Note On* et *Note Off* ; cela signifie que nous laissons de côté les

contrôles continus, ce qui ne nous empêche en aucun cas de les transporter sans les modifier.

Une STAN ne représente pas les évènements *Note On* et *Note Off* du SMF en tant que tels, mais les interprète et les fusionne afin de créer un unique évènement de *Note*, qui est temporellement défini par une position de début, une position de fin, et une longueur. Toutes les données d'un évènement de *Note* peuvent être décrites dans une *Event List* (« liste d'évènements »)⁸⁴, qui récapitule sous la forme d'un tableau les positions de début et de fin de chaque évènement par rapport à l'origine de la *timeline*⁸⁵ de la session, son statut — ici, « *Note* » —, le canal MIDI qui lui est affecté, sa hauteur, sa vitesse, sa longueur, et enfin deux autres champs dont le contenu est laissé à la discrétion de l'utilisateur.

Toutes les informations présentées dans l'*Event List* sont déduites des messages *Note On* et *Note Off* : il nous semble judicieux d'en faire autant et de combiner les données brutes du SMF ainsi. Autrement dit, notre premier objectif est de créer une liste d'évènements similaire à celle proposée par une STAN après l'import d'un SMF. Cela signifie que nous recherchons à fusionner et à réorganiser les messages *Note On* et *Note Off* afin de définir un unique évènement par note, caractérisé dans un premier temps par une hauteur tonale, une position de départ, une position de fin, et une longueur.

84 La « liste d'évènements » répertorie « l'ensemble des évènements ou régions d'un projet. La *liste d'évènements* permet de manipuler directement et de façon très précise les évènements et régions, sous forme numérique. Il est également possible d'y ajouter différents types d'évènements. » — *Guide de l'utilisateur de Logic Pro X*, p. 1970.

85 La « *timeline* » d'une session, que l'on peut traduire par « frise chronologique » ou « chronologie », correspond à une règle graduée qui permet de renseigner la position d'un évènement dans la fenêtre d'arrangement. La graduation de la *timeline* est plurielle et variable : elle peut être en mesures, en échantillons, en TC, ou encore en minutes. Bien évidemment, il est possible de passer d'une graduation à l'autre.

2. Import d'un SMF dans *Sibelius*

Tristan Noon passe en revue les différentes options qui sont proposées à l'import d'un SMF dans *Sibelius*⁸⁶. Avant de procéder à l'importation, *Sibelius* ouvre une fenêtre divisée en deux onglets : un premier onglet « Fichier MIDI », et un second « Notation ».

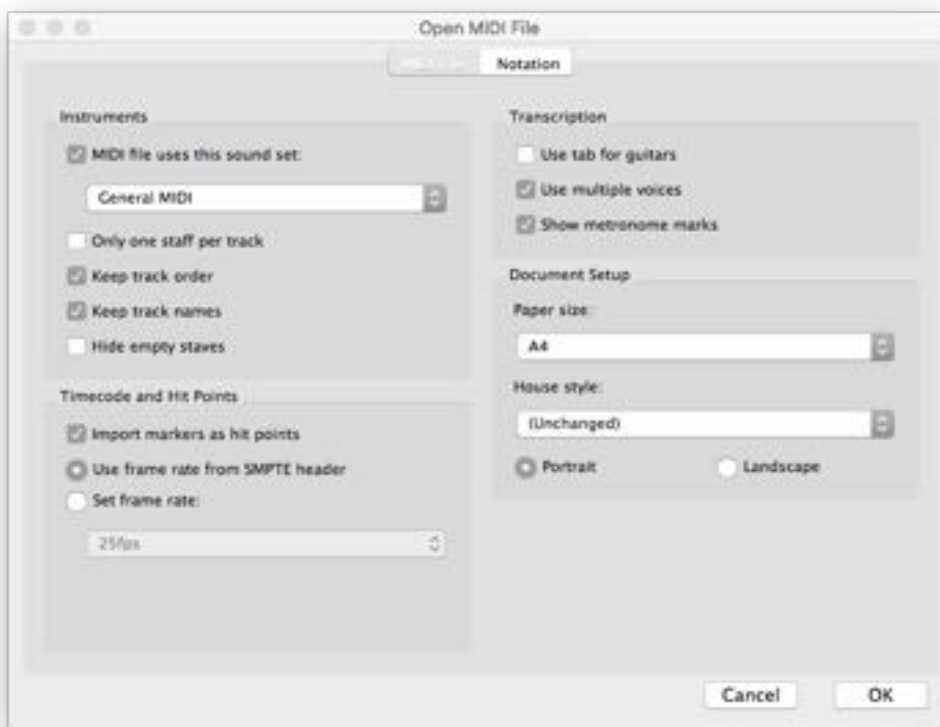


Figure 11 : Premier onglet de la fenêtre d'import d'un SMF dans *Sibelius*

L'onglet *Fichier MIDI* est subdivisé en quatre sous-onglets : « Instruments », « Timecode et Points de synchronisation », « Transcription » et « Organisation du document »⁸⁷. Le sous-onglet *Instruments* permet de choisir le « jeu de sons » avec lequel sera joué le fichier MIDI, mais aussi les

⁸⁶ Tristan Noon, *op. cit.*, pp. 55-59.

⁸⁷ Dans sa langue originale, c'est-à-dire en Anglais, l'onglet « *MIDI File* » (« Fichier *MIDI* ») contient les sous-onglets suivants : « *Instruments* », « *Timecode and Hit Points* », « *Transcription* » et « *Document Setup* ».

options suivantes : utiliser seulement une portée par piste, conserver ou non l'ordre et le nom des pistes, et enfin masquer ou non les portées vides après l'import du SMF. La second, *Timecode et Points de synchronisation*, offre la possibilité d'importer les marqueurs créés dans la STAN en tant que points de synchronisation, mais aussi de choisir si la fréquence-image à utiliser doit être trouvée dans le bloc d'en-tête, ou doit être fixée par *Sibelius*. Le sous-onglet *Transcription* propose de retranscrire les parties de guitare(s) sous la forme de tablatures, d'utiliser plusieurs voix, et de montrer ou non les signes métronomiques dans la partition. L'*Organisation du document*, enfin, présente quelques pré-réglages de mise en page, que nous ne détaillerons pas ici.

Si l'onglet *Fichier MIDI* ne doit pas être négligé, l'onglet *Notation* va davantage nous intéresser, puisqu'il conditionne l'interprétation et la transcription des données brutes du SMF faites par *Sibelius*. Autrement dit, la cohérence et la lisibilité de la partition résultante de l'import du SMF sont affinées grâce à cet onglet, qui se subdivise à son tour en quatre sous-onglets : « Valeurs de note », « Octaves du clavier », « Multiplets » et « Messages MIDI »⁸⁸.

Le premier onglet permet de faire des ajustements rythmiques en sélectionnant tout d'abord la plus petite valeur rythmique autorisée ou possible, de la ronde à la triple croche, puis de choisir de noter un *staccato* sur une note dont la longueur serait inférieure à un certain pourcentage de sa valeur minimum, ou un *tenuto* si cette dernière excédait cette longueur minimum d'un certain pourcentage ; par exemple, si on choisit de noter pour une longueur de note inférieure à 35 % un *staccato*, alors une note de longueur inférieure à 84 PPQN⁸⁹ sera notée comme une double-croche *staccato*. Une dernière option dans le sous-onglet *Valeurs de note* propose de retirer les soupirs entre les notes sur les portées de batterie. Le second onglet,

88 Toujours en langue anglaise, l'onglet « *Notation* » est subdivisé comme suit : « *Note Values* », « *Keyboard Staves* », « *Tuplets* » et « *MIDI Messages* ».

89 Si la résolution du SMF est égale à 960 PPQN, alors la longueur d'une double-croche vaut 240 PPQN. Dans ce cas, 35 % de 240 PPQN est égal à 84 PPQN.

Octaves du clavier, offre la possibilité de fixer automatiquement ou manuellement le point de séparation entre la portée inférieure et la portée supérieure du clavier⁹⁰. L'onglet *Triolets* autorise ou non la notation de certains multiplets : pour chacun d'eux, quatre options sont proposées à l'utilisateur, à savoir « *None* » (« aucun »), « *Simple* », « *Moderate* » (« modéré ») ou « *Complex* » (« complexe »). Ces quatre options affinent l'interprétation faite par *Sibelius* des informations du SMF, et évitent que leur transcription soit complexifiée inutilement. Enfin, l'onglet *Messages MIDI* propose de conserver ou non les différents types de messages MIDI contenus dans le SMF.

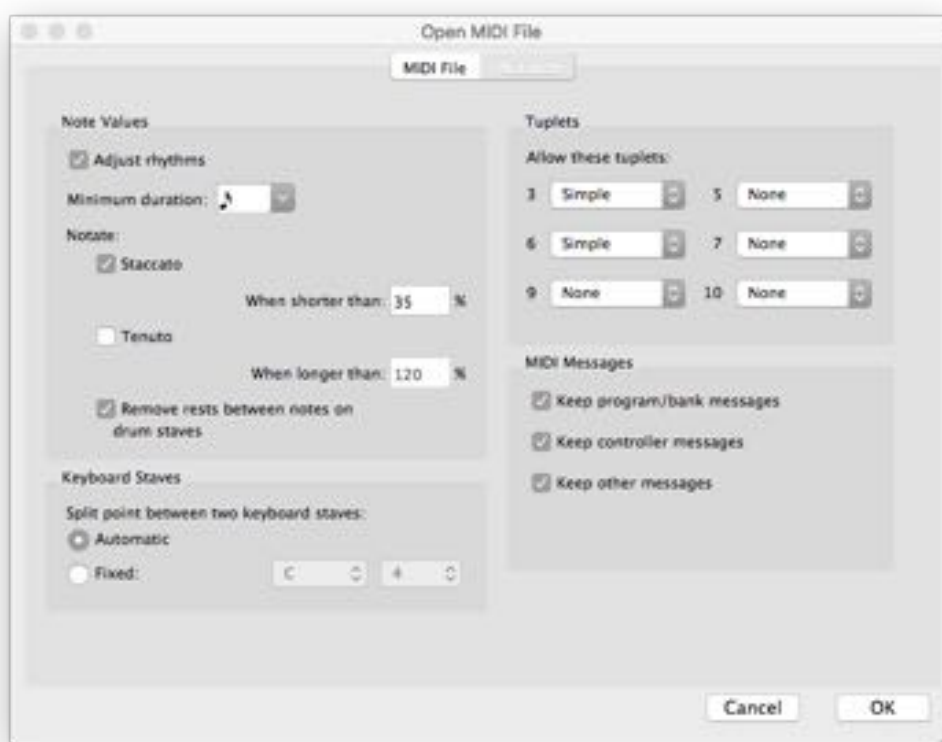


Figure 12 : Second onglet de la fenêtre d'import d'un SMF dans *Sibelius*

⁹⁰ La tessiture des instruments à claviers étant particulièrement étendue, comme pour le piano, il est nécessaire de les écrire sur plusieurs portées pour plus de lisibilité. En général, on les notes sur deux, avec la portée inférieure en clé de fa pour le registre grave de l'instrument, et la portée supérieure en clé de sol pour le registre aigu. Par ailleurs, nous rappelons qu'un lexique et quelques définitions des termes musicaux les plus utilisés peuvent être consultés en *Annexe 1*.

3. Conservation d'un socle commun de développement

Le principe de la liste d'évènements et les options d'importation d'un SMF dans *Sibelius* constituent un socle de développement solide sur lequel fonder notre outil logiciel d'interfaçage entre une STAN et la partition. En effet, la liste d'évènements nous permet de raisonner à la fois par le biais de positions et de longueurs temporelles absolues, qui induisent une flexibilité et des possibilités de réorganisation des données accrues, mais aussi par une appréhension plus globale et plus musicale des données de notes. La liste d'évènements ne présente qu'un seul item par note, dans lequel sont incluses toutes les données qui la définissent (positions, hauteur, vélocité, canal et durée). En d'autres termes, la liste d'évènements ordonne de manière claire et pratique les données que nous allons devoir analyser, et le cas échéant modifier.

Les options d'import d'un SMF dans *Sibelius*, quant à elle, sont à la fois une base de réflexion intéressante, mais aussi un inventaire des choix qui peuvent — ou doivent — être laissés à l'utilisateur. Le fait que l'utilisateur puisse choisir la plus petite valeur rythmique présente dans le SMF, ainsi que les différentes combinaisons de multiplats autorisées optimise l'adaptation des données brutes issues de la STAN, avant de les importer dans *Sibelius*. Toutefois, ces informations fournies par l'utilisateur sont dans notre cas utiles à une première phase d'analyse seulement : le principal manquement de *Sibelius* en matière d'import de données MIDI est d'appliquer ses paramètres de quantification et de transcription à l'ensemble du SMF, sans prendre en compte son contenu à un plus bas niveau de définition.

Nous proposons donc une analyse plus poussée et plus segmentée du SMF, afin d'adapter les paramètres de quantification et de transcription prédéfinis par l'utilisateur à chaque mesure — et pourquoi pas à chaque temps musical — dans le but de bannir un certain nombre d'incohérences, voire d'aberrations de transcription.

B. Proposition d'un cahier des charges fonctionnel

Le principe de notre outil logiciel est basé sur la comparaison de données MIDI brutes avec des tables de référence. Cette comparaison à plusieurs niveaux, associée à un certain nombre d'informations complémentaires, comme par exemple le nombre de notes présentes dans une mesure, devrait nous permettre de quantifier tous les évènements de *Notes* du SMF. Dans le cadre du développement de notre prototype, nous allons nous limiter à un SMF de *Format 1* contenant une seule piste musicale, soit un total de deux blocs de pistes. Dans un soucis de simplification, nous ne traiterons que le cas d'un instrument monophonique, et nous ne prendrons pas en compte les variations de signature rythmique.

Notre objectif est d'orienter, grâce à des informations connues, données par l'utilisateur, ou déduites, le choix de la table de référence la plus appropriée à confronter aux données brutes du SMF. Plusieurs phases d'analyse doivent nous permettre de répartir chaque évènement par mesure, puis par temps musical. Ce recensement de la répartition et de la longueur des données doit théoriquement nous permettre de choisir la table de référence adéquate. Ensuite, chaque donnée utile est confrontée à la table de comparaison choisie. Nous travaillerons dans un premier temps sur la quantification de la position de début de chaque note, puis dans un second temps sur celle de sa longueur. Il faudra enfin vérifier la cohérence des données quantifiées, avant de déconstruire la liste d'évènements obtenue après quantification, afin de fabriquer un nouveau SMF qui pourra être importé dans *Sibelius*.

En fin de compte, nous souhaitons définir et appliquer un pas de quantification qui s'adapte, suivant divers critères que nous définirons, aux données brutes du SMF. Pour le dire autrement, nous déterminons toutes les combinaisons et toutes les solutions possibles, afin de les restreindre pour chaque évènement jusqu'à ce qu'il ne reste plus qu'une seule possibilité de

quantification, c'est-à-dire une position de début et une longueur « évidentes ». Nous nous proposons donc de dresser un cahier des charges fonctionnel de cet outil logiciel, afin d'en préparer la programmation et le développement sous la forme d'un prototype.

1. Informations connues, données, ou déduites

Dans un premier temps, il est nécessaire de faire l'inventaire des informations que nous connaissons, c'est-à-dire de celles que nous pouvons immédiatement extraire du SMF, mais aussi de celles données par l'utilisateur. Nous déduirons de ces deux sources de données d'autres éléments qui nous seront utiles, comme par exemple la longueur d'une note, etc.

1.1. Informations globales issues du SMF

Précédemment, nous avons expliqué que toutes les données globales d'un SMF, notamment son format, son nombre de pistes, sa division et sa résolution temporelles, sont données dans son bloc d'en-tête ; dans notre cas, nous ne traitons que de SMF de *Format 1* dont la division temporelle est musicale, ce qui implique que $1 \text{ tick} = 1 \text{ PPQN}$. En général, la résolution d'un SMF est fixée à $R = 960 \text{ PPQN}$. Par ailleurs, pour un SMF de *Format 1*, tous les évènements méta qui ne sont pas propres à un canal ou une piste en particulier, comme les informations de synchronisation, sont rassemblés dans sa première piste. La première piste d'un SMF de *Format 1* ne contient donc pas de données de notes, et doit rester dédiée à ces évènements méta globaux. Ces derniers nous renseignent notamment sur la présence de marqueurs, mais aussi sur le *tempo* initial, la signature rythmique initiale, l'armure initiale et leurs éventuelles variations. L'ensemble des informations immédiatement fournies par le SMF sont récapitulées dans le tableau ci-après.

Préfixe/Type	Information
<i>BLOC D'EN-TÊTE</i>	
0x08	Format
0x10	Nombre de pistes
0x12	Division (Musicale/TC) et résolution <i>R</i> temporelles
<i>PISTE 1 (Format 1)</i>	
0xFF/0x06	Marqueur(s)
0xFF/0x51	Tempo (en microsecondes par noire)
0xFF/0x58	Signature rythmique
0xFF/0x59	Armure

Figure 13 : Informations globales issues du SMF

1.2. Informations données par l'utilisateur

Afin de fluidifier et d'optimiser l'analyse et la modification du SMF, nous avons fait le choix de demander un certain nombre d'informations à l'utilisateur. A l'instar de *Sibelius*, ce dernier est appelé à nous renseigner sur la plus petite valeur rythmique possible, ou autorisée (notée *NoteMin*), ainsi que sur la présence ou non de certains multiplats. Nous lui demandons également de nous indiquer la présence ou non de valeurs rythmiques pointées et/ou doublement pointées, ainsi que le nombre total de mesures contenues dans le SMF. L'ensemble des informations fournies par l'utilisateur sont récapitulées dans le tableau qui suit.

Information demandée	Nature de la réponse
Plus petite valeur rythmique (<i>noteMin</i>)	Un seul choix possible parmi cette liste : - ronde ; - blanche ; - noire ; - croche ; - double-croche ; - triple-croche.
Multiplets	(3) : oui/non ; (5) : oui/non ; (6) : oui/non ; (7) : oui/non ; (9) : oui/non ; (10) : oui/non.
Valeurs rythmiques pointées	Oui/non
Valeurs rythmiques double-pointées	Oui/non
Nombre total de mesures	Nombre

Figure 14 : Informations fournies par l'utilisateur

1.3. Informations déduites

Les informations directement extraites du SMF et celles fournies par l'utilisateur nous permettent d'en déduire d'autres, qui nous seront utiles pour déterminer le système de numération rythmique du SMF, mais surtout pour créer une liste d'évènements de *Notes*, ainsi qu'un certain nombre de tables de référence et de comparaison. La déduction et l'organisation de ces informations marquent la dernière étape de préparation avant l'analyse des données brutes extraites du SMF.

a. Détermination du système de numération rythmique

Afin de simplifier l'analyse des évènements de *Notes* à l'échelle du temps musical, il est préférable de différencier le système de numération binaire du système de numération ternaire. Comme sa dénomination l'indique, le système binaire fonctionne en base-2 : en musique, cela se traduit par le fait que chaque temps musical est divisible en deux parties égales ; par ailleurs, en

binaire, on utilise des valeurs rythmiques simples — et donc des mesures simples — pour représenter un temps musical. Dans le système ternaire, par contre, chaque temps peut se diviser en trois parties égales, et contrairement au système binaire, on utilise des valeurs pointées pour représenter un temps musical, et donc des mesures composées. Si une mesure est chiffrée 6/8, cela signifie qu'elle est composée de six croches, mais en ternaire, on préfère grouper les croches par trois pour constituer deux temps, chacun égal à une noire pointée : une mesure chiffrée 6/8 est donc composée de deux temps ternaires. Si le tempo est relativement lent, il est possible de battre la mesure « à la croche », ce qui revient à la décomposer en six croches, et donc à la rendre binaire, puisqu'une croche est toujours divisible en deux double-croches. Autrement dit, le système de numération rythmique n'influence que la division du temps musical et donc de la mesure, mais pas les rapports établis entre les valeurs rythmiques : une noire sera toujours égale à deux croches.

Quoiqu'il en soit, une STAN fonctionne uniquement en valeurs simples, ce qui signifie qu'elle « bat » une mesure ternaire à la croche, bien que son utilisateur l'appréhende à la noire pointée. Comme l'une de nos principales phases d'analyse se situe à l'échelle du temps et selon des critères musicaux, il est nécessaire de savoir si le système de numération rythmique du SMF est binaire ou ternaire. Cette information, qui sera renseignée textuellement, peut être déduite de la signature rythmique : si son dénominateur (*denR*) est égal à 2 ou à 4, alors la mesure est binaire ; au contraire, si le dénominateur est égal à 8 ou à 16, alors la mesure est ternaire. Notons tout de même deux cas particuliers : si la signature rythmique est égale à 3/8, alors la mesure est considérée binaire, sauf si elle est battue « à la mesure » ; si elle est égale à 6/4, alors elle est considérée comme étant théoriquement ternaire, mais compte-tenu du fonctionnement d'une STAN, nous considérerons une mesure à 6/4 comme une mesure binaire, c'est-à-dire que nous la décomposerons en six noires au lieu de deux blanches pointées.

Toutefois, dans le cadre du prototypage de notre outil, nous ne prendrons en compte que le cas des mesures binaires. Cela signifie qu'il faudra adapter un certain nombre de paramètres et de nos conclusions, afin d'ouvrir les fonctionnalités que nous allons proposer au système de numération ternaire. Pour la fabrication de notre liste d'évènements et de nos tables de référence, nous renseignerons donc la variable *Système rythmique* par la valeur *binary* (pour « binaire ») ou *ternary* (pour « ternaire »), qui permettra de déterminer le nombre de notes possibles par temps pour chaque valeur rythmique.

b. Création d'une liste d'évènements

Comme nous l'avons évoqué plus tôt, il est tout à fait possible de dresser une liste d'évènements d'après les données brutes extraites du SMF. En effet, les messages *Note On* et *Note Off* d'un SMF se présentent sous la forme d'une liste. Pour chaque évènement MIDI de *Note*, qu'il s'agisse indifféremment d'un message *Note On* ou *Note Off*, cinq valeurs sont renseignées : son Δt , son type, son canal et deux valeurs de données ; la première correspond à la hauteur tonale de la note et la seconde à sa vitesse.

Pour commencer, nous devons recalculer le Δt de chaque message pour le remplacer par un Δt_{abs} , qui est égal au délai entre l'origine temporelle du SMF et le déclenchement du message. Pour cela, il suffit d'ajouter le Δt_n d'un évènement MIDI n avec la somme des Δt des évènements qui le précèdent. Cela est d'autant plus nécessaire à la réorganisation et à la fusion des messages *Note On* et *Note Off* que dans un SMF, les messages sont ordonnés chronologiquement : par conséquent, il n'est pas rare que les messages *Note On* et *Note Off* pour une même note ne soient pas consécutifs, notamment lorsque le *tempo* est rapide, et/ou dans le cas d'une séquence de notes de courte durée. Une fois que le Δt_{abs} est calculé pour chaque évènement, les données brutes du SMF peuvent être réorganisées de telle sorte que pour une même note, le message *Note On* soit immédiatement suivi

par le message *Note Off*. Le fait que la position temporelle de chaque évènement soit désormais absolue, et non plus relative, rend possible une manipulation illimitée des données, ce qui était impossible auparavant, puisque les manipuler aurait nécessairement induit une perte de leur position et de leur synchronisation temporelles.

Event (1) : $\Delta t = 0$ type = 9 channel = 0 data = 60 ; 104	>>>>>>	Event (1) : $\Delta t_{abs} = 0$ type = 9 channel = 0 data = 60 ; 104	>>>>>>	Event (1) : $\Delta t_{abs} = 0$ type = 9 channel = 0 data = 60 ; 104
Event (2) : $\Delta t = 488$ type = 8 channel = 0 data = 60 ; 0	>>>>>>	Event (2) : $\Delta t_{abs} = 0 + 488 = 488$ type = 8 channel = 0 data = 60 ; 0	>>>>>>	Event (2) : $\Delta t_{abs} = 488$ type = 8 channel = 0 data = 60 ; 0
Event (3) : $\Delta t = 23$ type = 9 channel = 0 data = 62 ; 95	>>>>>>	Event (3) : $\Delta t_{abs} = 488 + 23 = 511$ type = 9 channel = 0 data = 62 ; 95	>>>>>>	Event (3) : $\Delta t_{abs} = 511$ type = 9 channel = 0 data = 62 ; 95
Event (4) : $\Delta t = 360$ type = 9 channel = 0 data = 64 ; 84	>>>>>>	Event (4) : $\Delta t_{abs} = 511 + 360 = 871$ type = 9 channel = 0 data = 64 ; 84	> > > > > > > > > > > > >	Event (5) : $\Delta t_{abs} = 985$ type = 8 channel = 0 data = 62 ; 0
Event (5) : $\Delta t = 114$ type = 8 channel = 0 data = 62 ; 0	>>>>>>	Event (5) : $\Delta t_{abs} = 871 + 114 = 985$ type = 8 channel = 0 data = 62 ; 0	> > > > > >	Event (4) : $\Delta t_{abs} = 871$ type = 9 channel = 0 data = 64 ; 84
Event (6) : $\Delta t = 894$ type = 8 channel = 0 data = 64 ; 0	>>>>>>	Event (6) : $\Delta t_{abs} = 985 + 894 = 1970$ type = 8 channel = 0 data = 64 ; 0	>>>>>>	Event (6) : $\Delta t_{abs} = 1970$ type = 8 channel = 0 data = 64 ; 0

Figure 15 : Exemple de calcul de Δt_{abs} et de réorganisation de six évènements MIDI

Une fois que les messages *Note On* et *Note Off* ont été réordonnés, ils peuvent être fusionnés au sein d'une liste d'évènements, qui sera organisée de la manière suivante :

Number	Type	Pitch	Velocity	Canal	Start	End	Length
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur

Figure 16 : Formatage d'une liste d'évènements

Bien entendu, la vitesse de l'évènement *Note Off* n'est pas prise en compte, puisqu'elle ne fait pas sens musicalement ; dans beaucoup de cas, elle est d'ailleurs égale à zéro. Le *Statut* de l'évènement est *Note*, sa vitesse est celle renseignée par le message *Note On*, tout comme le *Canal*, bien qu'il soit identique au canal emprunté par le message *Note Off*. Le *Début* n'est autre que le Δt_{abs} du message *Note On* et la *Fin* le Δt_{abs} du message *Note Off* correspondant à la note. Quant à la *Longueur*, elle est obtenue en soustrayant la valeur de *Début* à la valeur de *Fin*. Enfin, on numérote chaque évènement de 1 à x afin de les identifier et de les manipuler plus aisément.

c. Fabrication de tables de référence à l'échelle de la mesure et du temps musical

En parallèle de la fabrication de la liste d'évènements des données brutes du SMF, nous fabriquons deux tables de référence : une première à l'échelle de la mesure, et une seconde à l'échelle du temps musical. Nous connaissons le nombre de mesures qui sont incluses dans le SMF, et nous sommes en mesure de déduire le nombre de temps musicaux par mesure grâce à la signature rythmique. Le nombre de temps, ou de « battements », est la plus grande division de la mesure, et correspond au numérateur ($numR$) de la signature rythmique. Par exemple, pour une signature rythmique égale à 4/4, la mesure est divisée en quatre noires, tandis qu'en 12/8, la mesure est divisée en douze croches. Nous avons précédemment défini la résolution R d'un SMF ; cependant, cette résolution ne peut être utilisée que dans le cas où le dénominateur de la signature rythmique est égal à 4, puisque dans ce cas,

l'unité du temps musical correspond bien à la noire. Nous introduisons donc la résolution relative à l'unité de temps [musical] qui divise la mesure, notée r :

$$r = R \times \frac{denR}{4}$$

Ces deux informations, à savoir le nombre de mesures et la signature rythmique, associées à la résolution relative r , nous permettent de créer deux tables de référence de la longueur du SMF, soient une table de référence à la mesure, et une table de référence au temps musical.

La table de référence « à la mesure » indique le Δt_{abs} de chaque début de mesure. Sa longueur est égale au nombre de mesures indiqué par l'utilisateur, auquel on ajoute une mesure de « sécurité » : si l'utilisateur a indiqué que le SMF contenait douze mesures, alors la table de référence indique la position absolue de treize mesures. Par exemple, pour un SMF de résolution $r = 960$ PPQN contenant quatre mesures, avec une signature rythmique égale à 4/4, nous obtenons la table de référence suivante :

Mesure M_n	Δt_{abs} (PPQN)
1	$m_1 = 0$
2	$m_2 = m_1 + 4 \times 960 = 3840$
3	$m_3 = m_2 + 4 \times 960 = 7680$
4	$m_4 = m_3 + 4 \times 960 = 11520$
5	$m_5 = m_4 + 4 \times 960 = 15360$

Figure 17 : Exemple d'une table de référence à la mesure

Ainsi, nous pouvons généraliser le calcul de la position absolue m_n d'une mesure M_n , c'est-à-dire son Δt_{abs} , comme suit :

$$\begin{cases} m_1 = 0 \\ m_n = m_{n-1} + numR \times r \end{cases}$$

avec $numR$ le numérateur de la signature rythmique et r la résolution relative du SMF

De manière analogue, une table de référence « au temps musical » indique le Δt_{abs} de chaque temps de chaque mesure. Nous pouvons généraliser le calcul de la position absolue $b_{n,i}$ d'un temps musical $B_{n,i}$, avec n le numéro de la mesure à laquelle il appartient et i sont rang dans cette mesure — par exemple, le temps $B_{3,2}$ est le deuxième temps de la troisième mesure — comme suit :

$$\begin{cases} b_{1,n} = m_n \\ b_{i,n} = b_{i-1,n} + r \end{cases}$$

avec $1 \leq i \leq numR$ et r la résolution relative du SMF

Si nous reprenons l'exemple précédent, soit un SMF de résolution $r = 960$ PPQN et dont la signature rythmique est égale à 4/4, alors nous obtenons pour les deux premières mesures la table de référence « au temps musical » suivante :

Mesure - Temps	Δt_{abs} (PPQN)
1 - 1	$b_{1,1} = m_1 = 0$
1 - 2	$b_{1,2} = 0 + 960 = 960$
1 - 3	$b_{1,3} = 960 + 960 = 1920$
1 - 4	$b_{1,4} = 1920 + 960 = 2880$
2 - 1	$b_{2,1} = m_2 = 3840$
2 - 2	$b_{2,2} = 3840 + 960 = 4800$
2 - 3	$b_{2,3} = 4800 + 960 = 5760$
2 - 4	$b_{2,4} = 5760 + 960 = 6720$

Figure 18 : Exemple d'une table de référence au temps musical

Ces deux tables de référence nous permettront d'analyser notre liste d'évènements afin de répartir les évènements qu'elle contient d'abord par mesure, puis par temps musical.

d. Détermination et restriction des valeurs rythmiques possibles

Nous rappelons que la résolution temporelle R d'un SMF dans le cas d'une division temporelle musicale est relative à la noire, c'est-à-dire que la longueur de la noire est égale à la résolution R . Nous pouvons donc aisément calculer la longueur attendue des valeurs rythmiques « simples », qui sont données dans le tableau ci-dessous :

Valeur rythmique	Notation	Longueur (PPQN) (R=960 PPQN)
ronde	note1	$4 \times R = 3840$
blanche	note2	$2 \times R = 1920$
noire	note4	$R = 960$
croche	note8	$0,5 \times R = 480$
double-croche	note16	$0,25 \times R = 240$
triple-croche	note32	$0,125 \times R = 120$
quadruple-croche	note64	$0,0625 \times R = 60$
quintuple-croche	note128	$0,03125 \times R = 30$

Figure 19 : Calcul de la longueur des valeurs rythmiques « simples »

Nous pouvons aussi calculer la longueur des valeurs de notes pointées et doublement pointées d'après le calcul précédent. La longueur d'une note pointée est égale à une fois et demie la longueur de sa valeur non pointée, tandis que la valeur d'une note doublement pointée est égale à la somme de sa valeur simple, de la moitié et du quart de sa valeur. Par exemple, une noire pointée est égale à « 1,5 noires », ou à l'addition d'une noire et d'une croche. La longueur d'une noire doublement pointée est égale à « 1,75 noires », ou à une noire à laquelle on additionne la durée d'une croche et d'une double-croche. La longueur des valeurs rythmiques pointées et doublement pointées que nous pourrions rencontrer sont présentées dans les deux tableaux ci-après.

Valeur rythmique	Notation	Longueur (PPQN) (R=960 PPQN)
ronde pointée	note1Dot	$note1 + 0,5 \times note1 = 6 \times R = 5760$
blanche pointée	note2Dot	$3 \times R = 2880$
noire pointée	note4Dot	$1,5 \times R = 1440$
croche pointée	note8Dot	$0,75 \times R = 720$
double-croche pointée	note16Dot	$0,375 \times R = 360$
triple-croche pointée	note32Dot	$0,1875 \times R = 180$
quadruple-croche pointée	note64Dot	$0,09375 \times R = 90$
quintuple-croche pointée	note128Dot	$0,046875 \times R = 45$

Figure 20 : Calcul de la longueur des valeurs rythmiques pointées

Valeur rythmique	Notation	Longueur (PPQN) (R=960 PPQN)
ronde double-pointée	note1DotDot	$note1 + 0,75 \times note1 = 7 \times R = 6720$
blanche double-pointée	note2DotDot	$3,5 \times R = 3360$
noire double-pointée	note4DotDot	$1,75 \times R = 1680$
croche double-pointée	note8DotDot	$0,875 \times R = 840$
double-croche double-pointée	note16DotDot	$0,4375 \times R = 420$
triple-croche double-pointée	note32DotDot	$0,21875 \times R = 210$
quadruple-croche double-pointée	note64DotDot	$0,109375 \times R = 105$
quintuple-croche double-pointée	note128DotDot	$0,0546875 \times R = 52,5 \approx 52$

Figure 21 : Calcul de la longueur des valeurs rythmiques doublement pointées

Ensuite, nous calculons la longueur unitaire d'un certain nombre de multipliants. Nous entendons par « valeur unitaire » la longueur d'un multipliant divisée par le nombre de notes qui le composent : par exemple, pour un triolet de croches, nous calculons la longueur d'une croche de ce triolet, et non pas la longueur du triolet, qui en l'occurrence, est égale à la longueur d'une noire. Comme précédemment, on donne la longueur unitaire des multipliants que nous pourrions croiser dans le tableau ci-après.

Valeur rythmique	Notation	Longueur (PPQN) (R=960 PPQN)
triolet de ronde	note3	$1,33333 \times R = 1279,9968 \approx 1280$
triolet de blanche	note6	$0,66667 \times R = 640,0032 \approx 640$
triolet de noire	note12	$0,33333 \times R = 319,9968 \approx 320$
triolet de croche	note24	$0,16667 \times R = 160,0032 \approx 160$
triolet de double-croche	note48	$0,08333 \times R = 79,9968 \approx 80$
triolet de triple-croche	note96	$0,04167 \times R = 40,0032 \approx 40$
Triolet de quadruple-croche	note192	$0,02083 \times R = 19,9968 \approx 20$

Figure 22 : Calcul de la longueur unitaire des multiplats usuels

Ensuite, le système de numération rythmique que nous avons déterminé précédemment nous permet de calculer le nombre maximal d'évènements possibles par temps en fonction de la longueur de chaque valeur rythmique simple. En binaire, nous obtenons la table suivante :

Valeur rythmique	Notation	Nombre par temps
noire	note4	1
croche	note8	2
double-croche	note16	4
triple-croche	note32	8
quadruple-croche	note64	16
quintuple-croche	note128	32

Figure 23 : Table de référence du nombre maximal de valeurs rythmiques simples par temps

Une fois que ces différentes tables de référence ont été établies, nous pouvons les confronter à la signature rythmique et aux informations fournies par l'utilisateur dans le but de créer une table qui ne contienne que les valeurs de note que nous pouvons effectivement trouver dans une mesure du SMF que nous souhaitons quantifier. La signature rythmique permet d'écartier toutes les valeurs qui dépassent la longueur d'une mesure. Dans le cas d'une signature rythmique de 3/4, par exemple, toutes les valeurs strictement supérieures à la blanche pointée sont écartées. De manière analogue, la plus petite valeur possible renseignée par l'utilisateur (*NoteMin*), induit une valeur-limite basse.

Par conséquent, toutes les valeurs qui sont strictement inférieures à *NoteMin* ne sont pas conservées. Si par exemple *NoteMin* = *note16*, alors toutes les valeurs inférieures à la double-croche ne sont pas prises en compte. Enfin, nous ne conservons que les valeurs de multiplet indiquées par l'utilisateur ; il en est de même pour les valeurs pointées et doublement pointées.

Comme nous l'avons évoqué en préambule, nous avons pris la décision de nous affranchir d'un certain nombre de cas : nous ne considérerons donc aucun multiplet et aucune valeur doublement pointée ; de même, nous ne travaillerons qu'avec des signatures rythmiques binaires.

Afin d'illustrer la constitution d'une table de référence des valeurs rythmiques, prenons l'exemple d'un SMF de résolution $R = 960$ PPQN et de signature rythmique 4/4 ; l'utilisateur choisit *NoteMin* = *note16* et indique qu'il n'y a aucun multiplet, et aucune valeur doublement pointée ; les valeurs pointées sont autorisées. La table résultante de la confrontation de ces informations avec les tables de référence des valeurs rythmiques établies précédemment est présentée ci-après.

Données utiles issues du SMF : R = 960 PPQN Signature rythmique = 4/4 Système rythmique = <i>binary</i>			
Données utiles fournies par l'utilisateur : NoteMin = note16 Multiplets : aucun Valeurs pointées : oui Valeurs double-pointées : non			
Valeur rythmique	Notation	Longueur (PPQN)	Nombre par temps
ronde	note1	3840	-
blanche pointée	note2Dot	2880	-
blanche	note2	1920	-
noire pointée	note4Dot	1440	-
noire	note4	960	1
croche pointée	note8Dot	720	-
croche	note8	480	2
double-croche pointée	note16Dot	360	-
double-croche	note16	240	4

Figure 24 : Exemple d'une table de référence des valeurs rythmiques possibles après recoupement avec les données fournies par l'utilisateur

Avec cette table de référence des valeurs rythmiques « possibles » ou « autorisées », nous sommes maintenant en possession de tous les éléments nécessaires à l'analyse des données brutes du SMF réorganisées dans la liste d'évènements que nous avons déjà décrite.

2. Création de sous-listes d'évènements

Cette première phase d'analyse se subdivise en deux étapes. Dans un premier temps, le but est de créer autant de sous-listes d'évènements qu'il y a de mesures, puis de répartir chaque évènement par temps musical au sein de ces sous-listes.

2.1. Répartition des évènements par mesure

Pour répartir les temps par mesure, nous comparons le Δt_{abs} de *Début* de chaque évènement avec notre table de référence « par mesure ». La plus petite valeur rythmique possible, *noteMin*, nous permet d'affiner l'assignation des cas-limites à telle ou telle mesure. Cette première répartition « par mesure » est exécutée grâce à l'algorithme suivant :

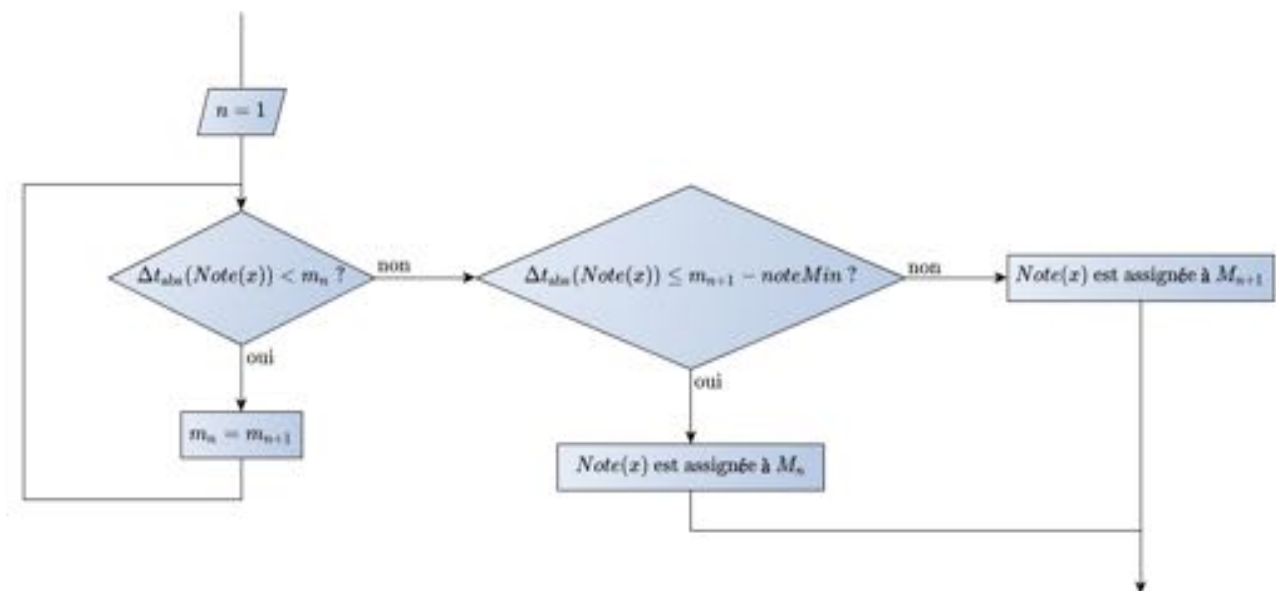


Figure 25 : Algorithme de répartition des évènements par mesure

2.2. Répartition des évènements par temps

Grâce à l'algorithme précédent, nous obtenons autant de sous-listes d'évènements que le SMF contient de mesures comprenant au moins un évènement de *Note*. Pour chacune de ces sous-listes, nous répartissons les évènements qu'elle contient par temps musical. Attention, les données brutes de chaque évènement ne sont pas encore modifiées : la répartition des évènements par mesure et par temps est une information supplémentaire associée à chaque évènement, sans pour autant le modifier ; il s'agit d'une

valeur purement indicative. De manière analogue à la répartition des évènements par mesure, nous comparons chaque sous-liste d'évènements avec la table de référence qui lui correspond grâce à l'algorithme suivant :

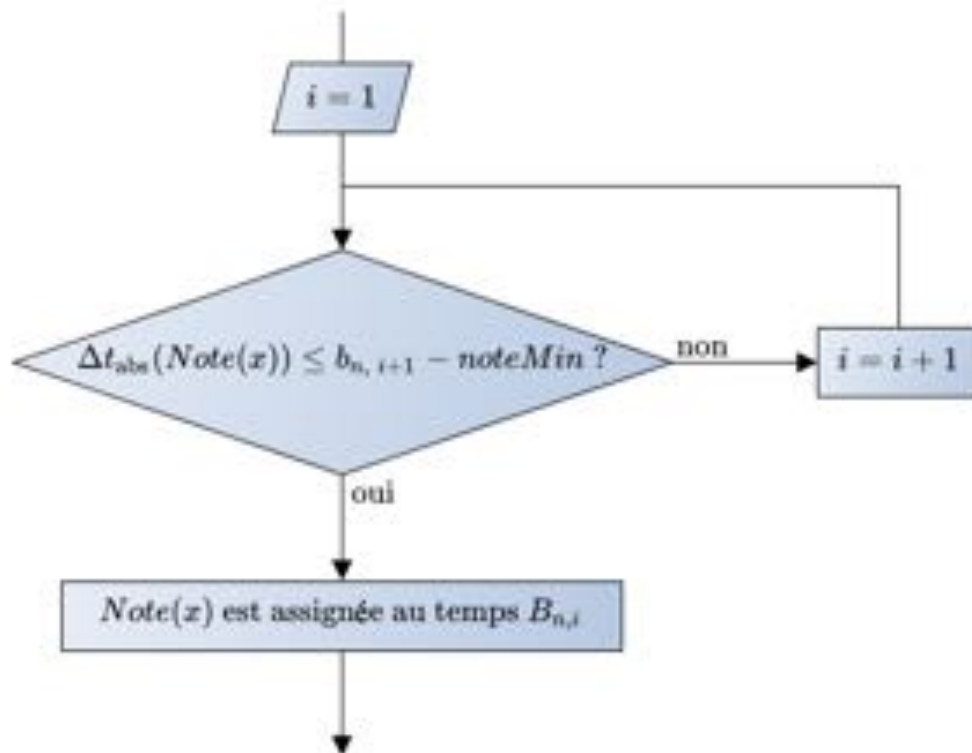


Figure 26 : Algorithme de répartition des évènements par temps

2.3. Calcul du nombre d'évènements par mesure et par temps

Une fois que chaque évènement a été associé à une mesure et à un temps de cette mesure, nous pouvons aisément calculer le nombre d'évènements par mesure et par temps, respectivement notés *nbrEventsBar* et *nbrEventsBeat*. En recoupant ces données avec celles qui sont déjà en notre possession, nous pouvons choisir le pas de quantification le plus approprié pour chaque temps musical. Notons que nous connaissons aussi le nombre d'évènements de *Notes* sur l'intégralité du SMF ; nous le notons *nbrEvents*.

L'exemple ci-dessous illustre une répartition possible des *Notes* et des données qui leur sont associées par mesure et par temps, avec une signature rythmique égale à 4/4 et une résolution $R = 960$ PPQN.

Position temporelle absolue de la mesure M_1 : $m_1 = 0$ PPQN Nombre d'évènements dans la mesure $M_1 = nbrEventsBar(1)$								
Temps musical $B_{1,1}$: $b_{1,1} = 0$ PPQN Nombre d'évènements compris dans le temps $B_{1,1} = nbrEventsBeat(b_{1,1})$								
Number	Type	Pitch	Velocity	Canal	Start	End	Length	Beat
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur	Temps
Temps musical $B_{1,2}$: $b_{1,2} = 960$ PPQN Nombre d'évènements compris dans le temps $B_{1,2} = nbrEventsBeat(b_{1,2})$								
Number	Type	Pitch	Velocity	Canal	Start	End	Length	Beat
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur	Temps
[...]								
Temps musical $B_{1,4}$: $b_{1,4} = 2880$ PPQN Nombre d'évènements compris dans le temps $B_{1,4} = nbrEventsBeat(b_{n,i})$								
Number	Type	Pitch	Velocity	Canal	Start	End	Length	Beat
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur	Temps
Position temporelle absolue de la mesure M_2 : $m_2 = 3840$ PPQN Nombre d'évènements dans la mesure $M_2 = nbrEventsBar(2)$								
Temps musical $B_{2,2}$: $b_{2,1} = 3840$ PPQN Nombre d'évènements compris dans le temps $b_{2,1} = nbrEventsBeat(b_{2,1})$								
Number	Type	Pitch	Velocity	Canal	Start	End	Length	Beat
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur	Temps
[...]								
Position temporelle absolue de la mesure M_n : $m_n = m_{n-1} + 3840$ PPQN Nombre d'évènements dans la mesure $M_n = nbrEventsBar(n)$								
Temps musical $B_{n,1}$: $b_{n,1} = m_n$ PPQN Nombre d'évènements compris dans le temps $B_{n,1} = nbrEventsBeat(b_{n,1})$								
Number	Type	Pitch	Velocity	Canal	Start	End	Length	Beat
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur	Temps
[...]								
Temps musical $B_{n,4}$: $b_{n,4} = b_{n,3} + 960$ PPQN Nombre d'évènements compris dans le temps $B_{n,4} = nbrEventsBeat(b_{n,4})$								
Number	Type	Pitch	Velocity	Canal	Start	End	Length	Beat
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur	Temps

Tableau 1 : Exemple de sous-listes d'évènements répartis par mesures et par temps, pour un SMF de résolution $R = 960$ PPQN et de signature rythmique 4/4

3. Quantification adaptative des données

Après ces étapes successives d'analyse, il est temps de traiter les données, c'est-à-dire de quantifier chaque évènement de *Note*. Cela revient à éditer deux composantes temporelles pour chacun d'eux, à savoir leur position de *Début*, et leur longueur. Toutefois, nous considérons que le nombre d'évènements par mesure influence le traitement de ces deux valeurs temporelles de manière similaire, puisqu'il nous permet de retirer la valeur la plus haute de notre table de référence. Autrement dit, si une mesure compte plus d'un évènement en son sein, alors elle ne peut pas contenir la valeur rythmique qui est égale à une mesure complète. Par contre, si elle contient plus d'un évènement, alors les combinaisons sont trop nombreuses et le nombre d'évènements présents dans la mesure trop vague pour pouvoir être utilisé pertinemment. En effet, si une mesure chiffrée 4/4 compte trois évènements, il peut tout aussi bien s'agir d'une blanche pointée et de deux croches, d'une noire pointée et de deux double-croches, ou encore de trois noires, etc. La mesure n'est pas une échelle assez précise pour aller plus loin que la première valeur de notre table de référence : pour cela, aussi bien pour le calcul de la position de *Début* que de la longueur d'une note, il nous faut descendre à l'échelle du temps musical.

3.1. Quantification de la position de début de chaque note

Dans un premier temps, nous nous consacrons uniquement à la quantification de la position de début de chaque *Note*. Cela signifie qu'une noire peut être alignée par rapport à la croche, sans pour autant que sa longueur ne soit raccourcie pour devenir une croche. Par exemple, si nous comptons deux évènements dans un temps, alors ils ne peuvent pas avoir la même position de *Début*, ce qui implique qu'ils ne peuvent pas être alignés par rapport à la noire. *Grosso modo* — nous développerons ensuite les algorithmes impliqués dans ces choix —, le *Début* de chacune de ces deux notes sera

quantifié par rapport à la croche ou à la double, ce qui n'empêchera pas de déterminer que le second évènement a la longueur d'une noire. La figure ci-dessous présente l'exemple d'un rythme de syncope sur deux temps, pour lequel il est nécessaire d'appliquer un pas de quantification égal à la croche pour déterminer la position de début de chaque *Note* ; chaque temps musical est représenté par un triangle et la seconde croche de chaque temps par une croix sur la ligne inférieure de la portée.

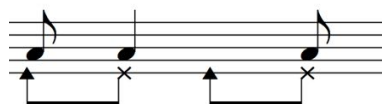


Figure 27 : Exemple de rythme de syncope sur deux temps

a. Réduction de la fenêtre d'analyse

Afin de réduire notre fenêtre d'analyse, il faut nous intéresser au nombre d'évènements inclus dans chaque temps musical. Logiquement, le fait de nous placer à l'échelle du temps musical, c'est-à-dire à l'échelle de la valeur rythmique unitaire, nous affranchit de toutes les valeurs de notre table qui lui sont strictement supérieures. Dans le cas d'une signature rythmique égale à 4/4, par exemple, nous pouvons exclure systématiquement toutes les valeurs strictement supérieures à la noire (*note4*). De plus, de la même façon que nous avons écarté la valeur la plus haute de notre table de référence pour toutes les mesures contenant plus d'un évènement, nous ne prenons pas en compte la valeur rythmique unitaire pour des temps comprenant au moins deux évènements. Enfin, comme les valeurs pointées et doublement pointées ne diffèrent des valeurs rythmiques simples que par leur longueur, nous les excluons également de la fenêtre d'analyse pour la quantification de la position de début de chaque note. En d'autres termes, nous ne conservons que les valeurs rythmiques simples qui n'ont pas été éliminées pour les raisons que nous avons déjà évoquées, et éventuellement les multiplats sélectionnés par l'utilisateur.

b. Choix d'un pas de quantification

Pour choisir le pas de quantification le plus approprié, nous comparons le nombre d'évènements compris effectivement dans un temps avec le nombre maximal de notes qu'un temps peut compter en fonction de chaque valeur rythmique unitaire. À chacune de ces quantités de notes par temps est associée une valeur de note, notée *noteGrid* — où *Grid* correspond à la division de la ronde par cette note ; par exemple, pour *note4*, *Grid* = 4. Enfin, à chaque *noteGrid* est associée une longueur, calculée précédemment ; elle est notée *length(noteGrid)*. Autrement dit, *length(noteGrid(d))* est égale à la longueur de la note correspondant à l'item *d* de la table de référence précédente. Le but de tout ceci est de créer une table de comparaison temporaire pour un temps $B_{n,i}$, dont le pas est égal à *length(noteGrid(d))*. L'algorithme ci-après présente la méthode de choix d'un pas de quantification pour un temps musical donné.

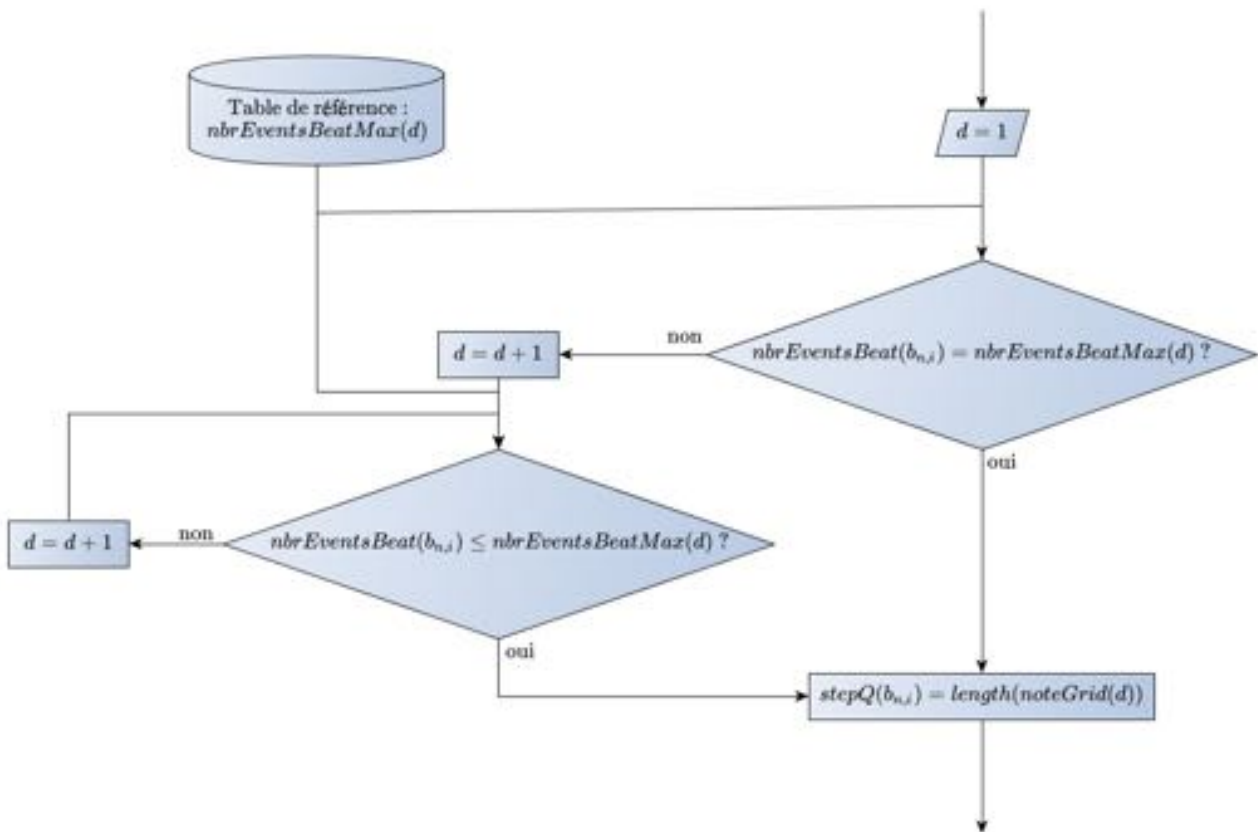


Figure 28 : Algorithme de choix d'un pas de quantification pour un temps musical donné

c. Analyse et modification des données brutes

Pour chaque temps musical, nous avons déterminé un pas de quantification $stepQ(b_{n,i})$. Grâce à ce pas de quantification, nous allons pouvoir créer une table de référence afin d'associer la position absolue de chaque évènement à la position absolue « attendue » la plus proche. L'algorithme suivant permet de créer une table de comparaison pour un temps musical et un pas de quantification donnés :

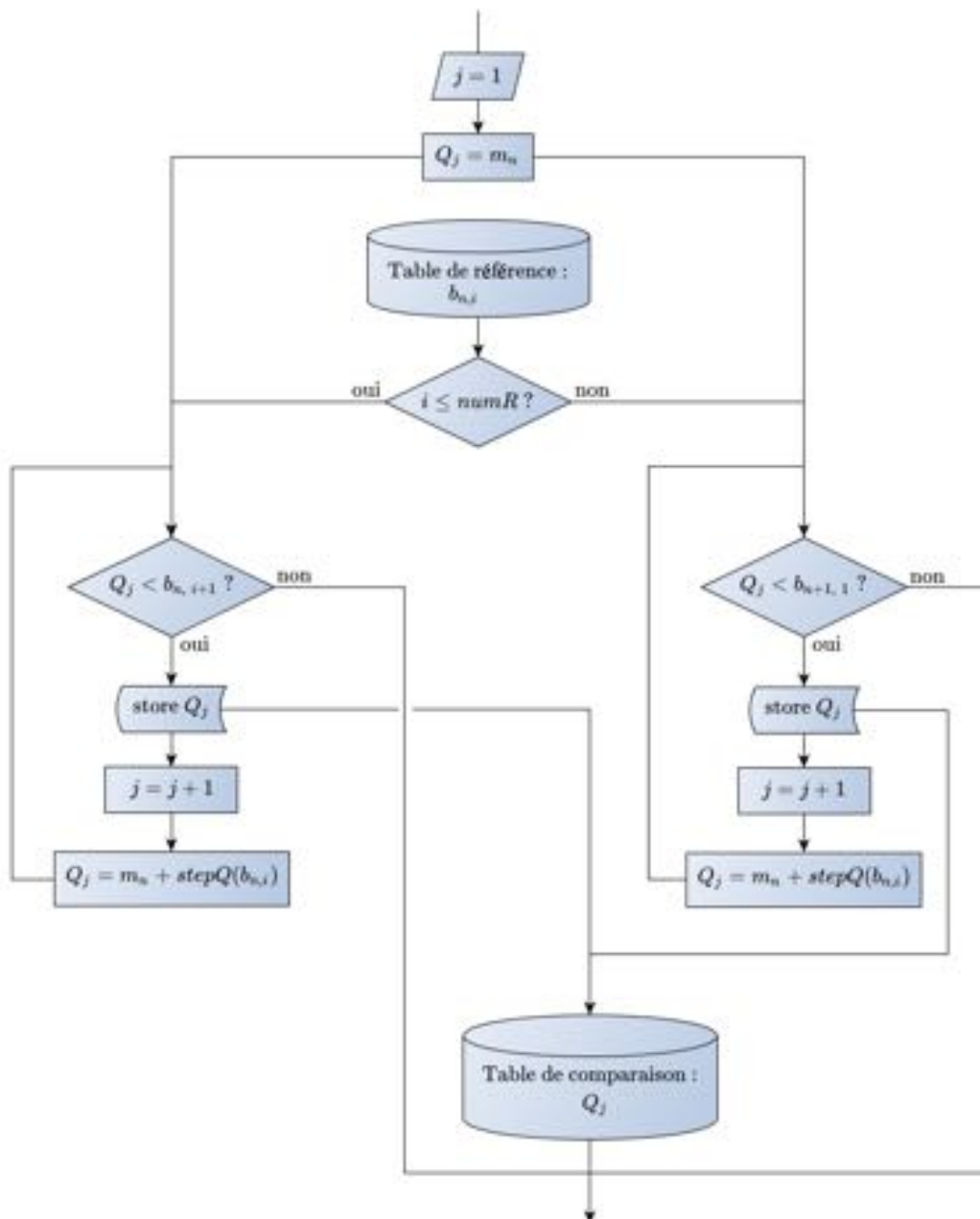


Figure 29 : Algorithme de fabrication d'une table de comparaison pour un temps et un pas de quantification donnés

Cette table de comparaison contient toutes les positions temporelles absolues possibles pour un pas de quantification donné, sur lesquelles nous alignons les positions brutes des évènements compris dans ce temps musical. Voici un exemple de table de comparaison pour $m_n = 3840$ PPQN et $stepQ(b_{2,3}) = 240$ PPQN :

Données utiles :	
$R = 960$ PPQN	
$b_{2,3} = 5760$ PPQN	
$stepQ(b_{2,3}) = 240$ PPQN	
Q_j	Δt_{abs} (PPQN)
Q ₁	5760
Q ₂	6000
Q ₃	6240
Q ₄	6480

Tableau 2 : Exemple d'une table de comparaison

Une fois que la table de quantification temporaire a été dressée, elle peut être comparée aux données MIDI grâce à l'algorithme suivant⁹¹ :

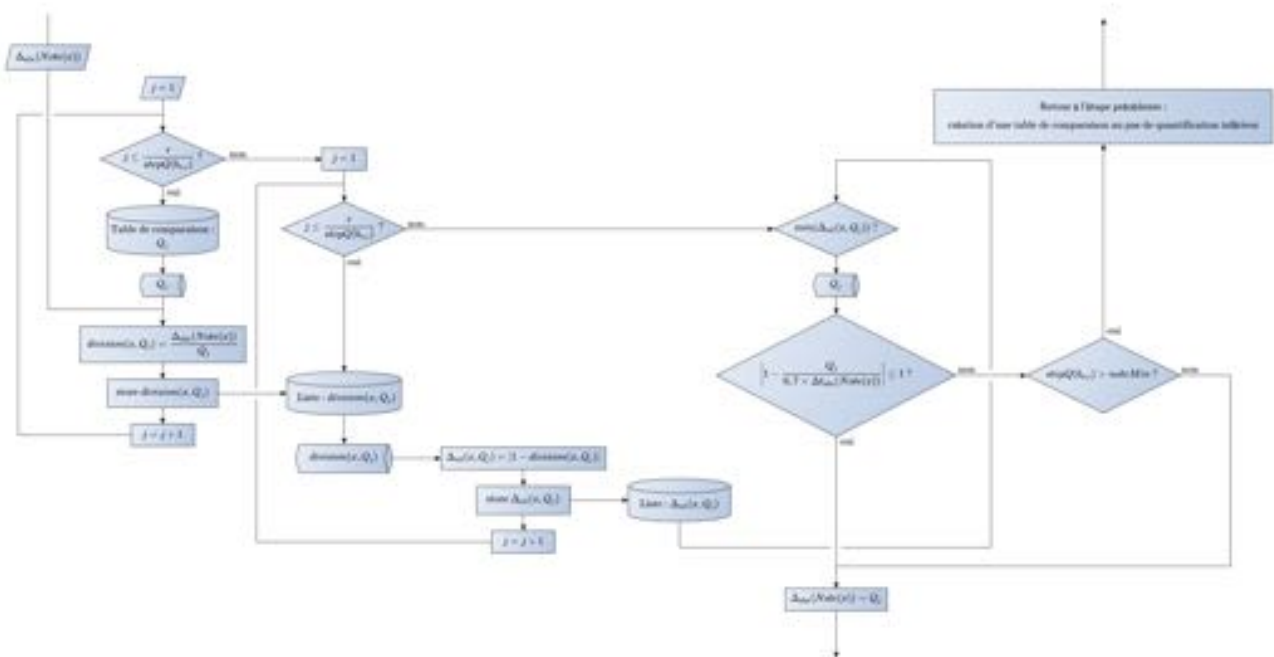


Figure 30 : Algorithme de quantification du Δt_{abs} d'une Note(x) donnée

91 L'algorithme de quantification du Δt_{abs} d'une Note(x) donnée (Figure 30), est reproduit à plus grande échelle en Annexe 4.

Pour une note donnée, nous divisons son Δt_{abs} par chaque position possible dans le temps, c'est-à-dire par toutes les valeurs de Q calculées au préalable. Ensuite, nous déterminons la valeur absolue de l'écart entre Δt_{abs} et chaque valeur de Q , pour enfin choisir la position qui présente le plus petit écart. Toutefois, nous estimons que la marge d'erreur par rapport à la position de départ « attendue » d'une note est de l'ordre de 20 % de la longueur de la plus petite valeur rythmique possible⁹². Par conséquent, si le pas de quantification utilisé n'est pas le pas de quantification le plus petit possible — autrement dit, si le pas de quantification choisi est supérieur à la longueur de *noteMin* —, et que $\Delta_{rel}(x, Q_j)$ est supérieur ou égal à 0,3 — cette valeur devra être précisée par l'expérience —, alors nous reproduisons le processus avec le pas de quantification inférieur, et nous quantifions $\Delta t_{abs}(x)$ avec la valeur de la table qui présente le plus petit écart par rapport à sa valeur brute.

3.2. Quantification de la longueur de chaque note

A présent que le Δt_{abs} de chaque note a été quantifié, il est temps de quantifier leur longueur. Rappelons tout d'abord que l'analyse du nombre d'évènements par mesure nous a permis d'exclure ou non la plus haute valeur rythmique possible de notre analyse. Pour tous les évènements dont la longueur est inférieure à la plus petite valeur rythmique possible, nous allongeons leur longueur pour qu'elle soit égale à cette dernière. De cette manière, notre fenêtre d'analyse est bornée, et toutes les *Notes* dont la longueur est strictement inférieure à *noteMin* sont quantifiées immédiatement.

Avant de présenter le fonctionnement général de l'algorithme qui permet de quantifier la longueur d'un évènement de *Note* quelconque, il nous faut aborder le cas des notes qui chevauchent une ou plusieurs mesure(s). Ce cas

⁹² Ce pourcentage a été déterminé dans le cadre d'expérimentations personnelles : il faut évidemment le préciser, et peut-être imaginer une personnalisation de ce paramètre par rapport au profil de chaque utilisateur. Il pourrait être intéressant de proposer à un nouvel utilisateur une courte série de tests afin de calibrer notre outil logiciel par rapport à son jeu.

particulier ne peut se présenter que pour le dernier évènement de chaque mesure. Dans ce cas, on calcule la longueur de la portion qui dépasse de la mesure où elle est jouée, et on la compare aux valeurs rythmiques de référence. Si cette part de chevauchement est significative, nous la quantifions puis nous l'additionnons avec la longueur de la portion de la mesure occupée par la note en question. Si la longueur de la portion de chevauchement est inférieure au pas de quantification choisi pour quantifier le début de la note, alors ce dépassement est considéré comme négligeable, et n'est pas pris en compte. Enfin, il est possible que cette note chevauche aussi la note qui lui succède : nous verrons comment palier à ce problème au moment de la vérification et de la correction éventuelle des données.

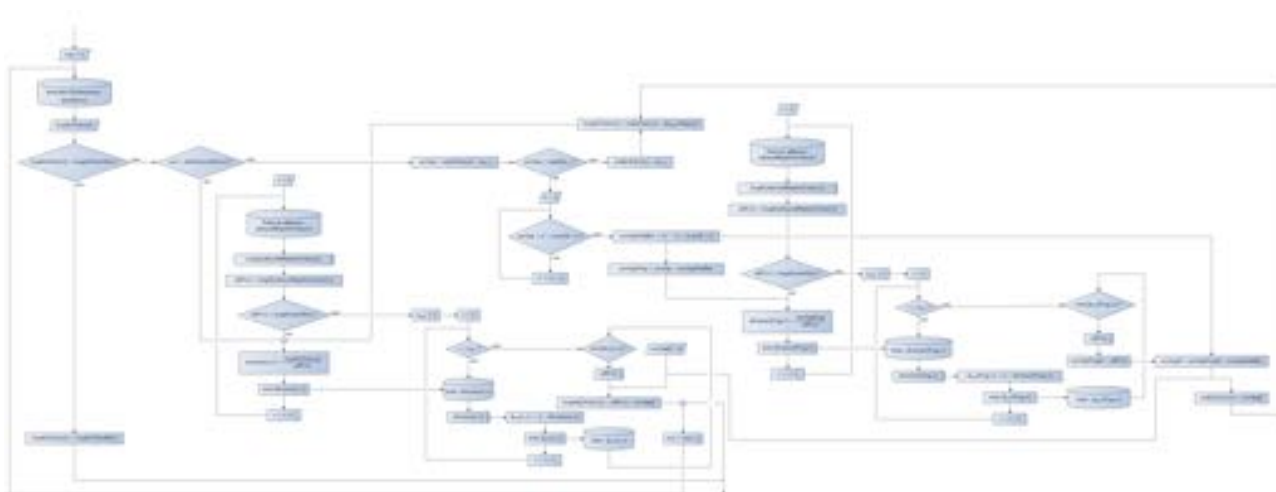


Figure 31 : Algorithme de quantification de la longueur d'une *Note(x)* donnée

L'algorithme de quantification de la longueur d'une *Note(x)*⁹³ est relativement similaire à celui de quantification de sa position de *Début*. Il compare la longueur $length(Note(x))$ de chaque note à toutes les valeurs rythmiques possibles, afin de déterminer quelle est celle dont elle se rapproche le plus. La quantification de $length(Note(x))$ revient alors tout simplement à remplacer sa valeur par la valeur rythmique de référence la plus proche.

⁹³ L'algorithme de quantification de la longueur d'une *Note(x)* donnée (Figure 31), est reproduit à plus grande échelle en Annexe 5.

3.3. Vérification et correction éventuelle des données

Une dernière phase d'analyse et d'édition des données est nécessaire pour corriger les éventuelles aberrations ou irrégularités qui auraient pu advenir lors de la quantification de la position temporelle absolue ou de la longueur de chaque note. A ce stade, nous sommes en possession d'une liste d'évènements qui contient les informations suivantes — la lettre « Q » est ajoutée à la fin de la notation de chaque valeur qui a été quantifiée :

Number	Type	Pitch	Velocity	Canal	Start	End	Length
Numéro	Type	Hauteur	Vélocité	Canal	Début	Fin	Longueur
1	NoteQ	pitch(1)	vel(1)	canal(1)	$\Delta t_{abs}Q(1)$	endQ(1)	lengthQ(1)
2	NoteQ	pitch(2)	vel(2)	canal(2)	$\Delta t_{abs}Q(2)$	endQ(2)	lengthQ(2)
3	NoteQ	pitch(3)	vel(3)	canal(3)	$\Delta t_{abs}Q(3)$	endQ(3)	lengthQ(3)
4	NoteQ	pitch(4)	vel(4)	canal(4)	$\Delta t_{abs}Q(4)$	endQ(4)	lengthQ(4)
[...]							
x	NoteQ	pitch(x)	vel(x)	canal(x)	$\Delta t_{abs}Q(x)$	endQ(x)	lengthQ(x)

Figure 32 : Liste d'évènements après quantification de la position temporelle absolue et de la longueur de chaque item

Un instrument monophonique ne peut pas jouer deux notes superposées, ce qui veut dire qu'une note ne peut pas être jouée tant que celle qui la précède n'est pas stoppée. La précision de la position d'une note non quantifiée étant plus précise que sa longueur, nous considérons que $\Delta t_{abs}Q$ est définitif. Il reste donc à vérifier que la fin d'une note n'empiète par sur celle qui suit, notamment dans le cas d'une note à cheval sur plusieurs mesures. Pour cela, nous soustrayons la position absolue de fin de la première note $endQ(x)$ — obtenue en soustrayant $\Delta t_{abs}Q(x)$ à la longueur $lengthQ(x)$ — à la position temporelle absolue $\Delta t_{abs}Q(x+1)$ de la note suivante. On vérifie que le résultat est supérieur ou égal à zéro. Si ce n'est pas le cas et que le résultat est strictement négatif, alors $lengthQ(x) > \Delta t_{abs}Q(x+1)$, donc la $Note(x)$ est trop longue et il faut donc la raccourcir, soit $lengthQ(x) = \Delta t_{abs}Q(x+1)$. L'ensemble de ce processus de vérification et de correction est détaillé dans le schéma ci-après.

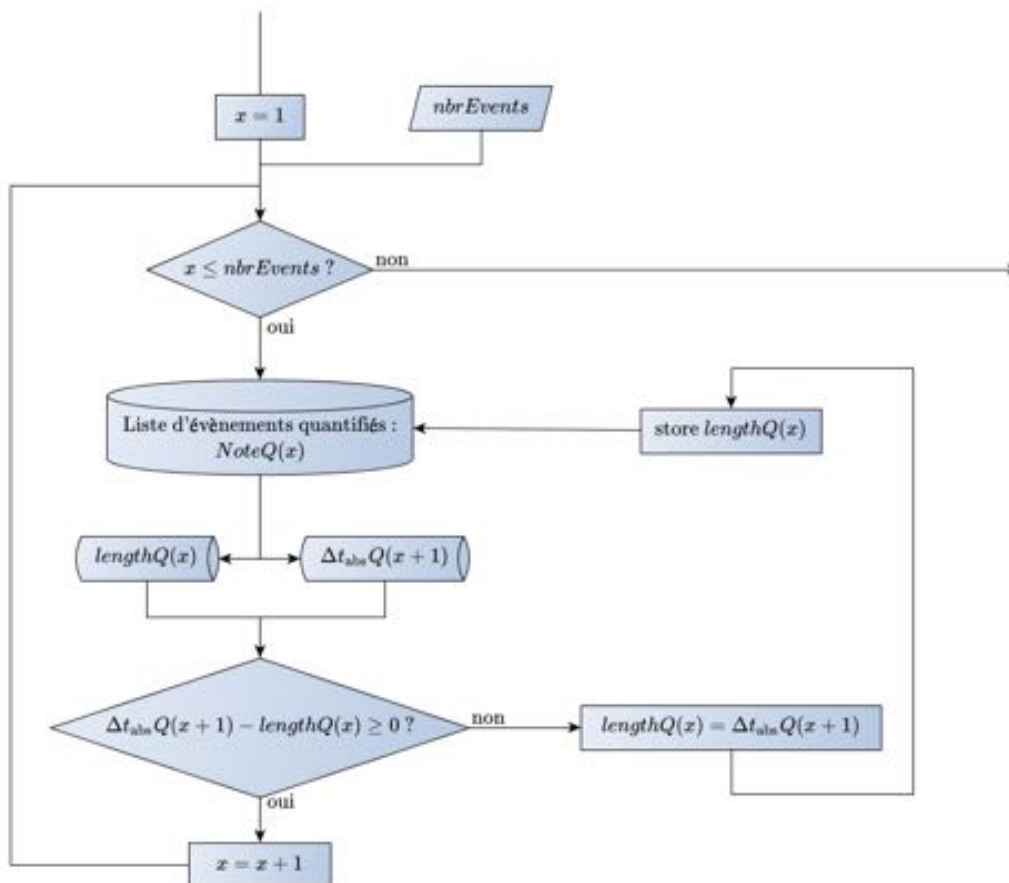


Figure 33 : Algorithme de vérification et de correction de la longueur quantifiée des évènements $Note(x)$ de la liste d'évènements

4. Reconstitution d'un SMF quantifié

Nous sommes enfin en possession d'une liste cohérente d'évènements quantifiés. Cependant, la représentation des données MIDI de cette liste ne correspond pas à la syntaxe attendue par un SMF, et il faut donc les réorganiser, afin de pouvoir reconstituer un SMF importable dans *Sibelius*. Pour cela, nous allons employer à rebours la méthode que nous avons mise en œuvre pour fabriquer la liste d'évènements initiale. Nous rappelons que l'évènement *Note* est défini par sept valeurs en-dehors de son type : son positionnement par rapport aux autres évènements, sa hauteur tonale, sa vélocité, son canal MIDI, sa position de début absolue, sa position de fin absolue, et enfin sa longueur. Or, un évènement MIDI standard est défini par

le Δt qui le sépare de l'évènement qui le précède, de son type, de son canal, et dans le cas d'un message *Note On* ou *Off*, de deux valeurs de donnée, qui définissent sa hauteur tonale et sa vélocité ; le message *Note On* est de type 9, tandis que le message *Note Off* est de type 8. Pour un instrument monophonique, les messages *Note On* et *Note Off* sont systématiquement dans l'ordre de déclenchement et d'arrêt de chaque note, puisque ces dernières ne peuvent pas se superposer. Nous donnons ci-dessous la méthode de transposition d'une liste d'évènements en messages *Note On* et *Note Off* :

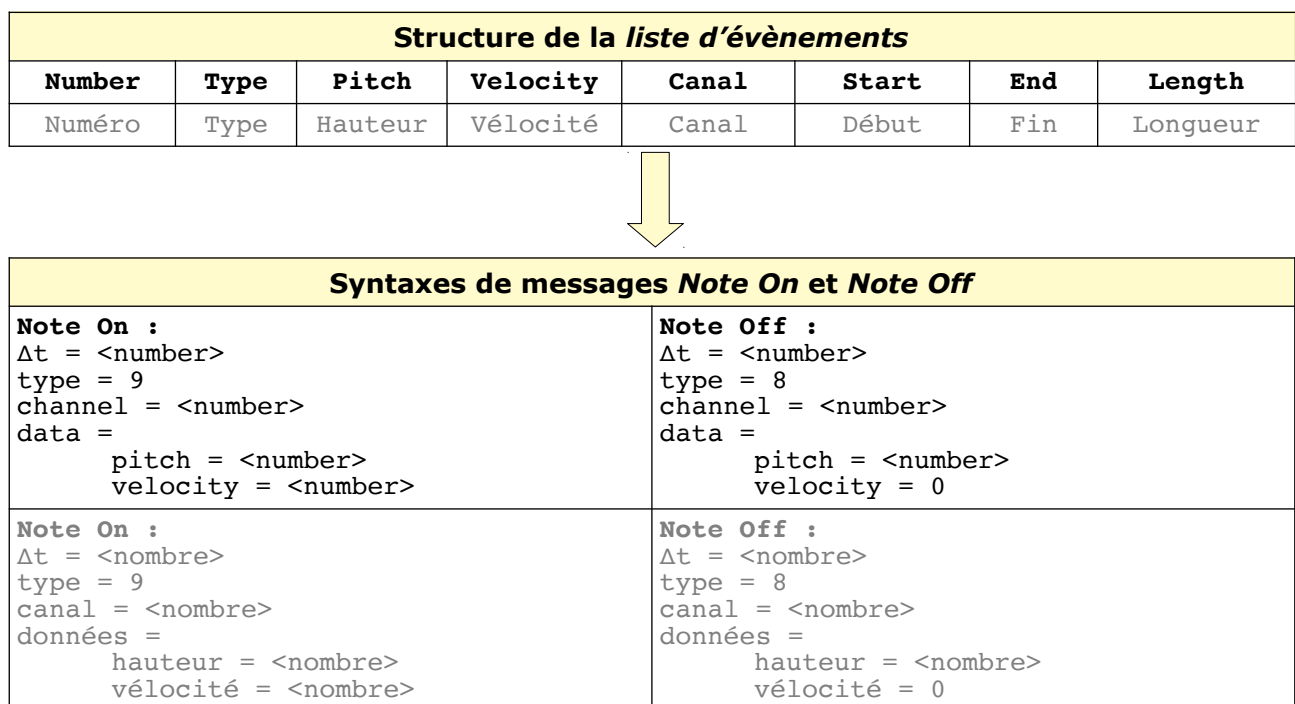


Figure 34 : Transposition d'une liste d'évènements vers des messages MIDI standards

Dans la mesure où nous n'avons pas ajouté ou supprimé d'évènements de notre SMF initial, nous pouvons nous contenter d'en modifier les valeurs. Pour cela, nous utilisons la liste d'évènements contenant les données brutes réorganisées que nous avons traitées, afin de modifier les informations qu'elle contient selon la méthode décrite ensuite.

On fixe les variables suivantes :

```
start(Note(-1)) = 0  
end(Note(-1)) = 0  
length(Note(-1)) = 0
```

Soit, pour un évènement Note(x) :

Note On (x) =

```
 $\Delta t = \text{start}(\text{Note}(x)) - \text{start}(\text{Note}(x-1)) - \text{length}(\text{Note}(x-1))$   
type = 9  
channel = channel(Note(x))  
data =  
    pitch = pitch(Note(x))  
    velocity = velocity(Note(x))
```

Note Off (x) =

```
 $\Delta t = \text{length}(\text{Note}(x))$   
type = 8  
channel = channel(Note(x))  
data =  
    pitch = pitch(Note(x))  
    velocity = 0
```

Figure 35 : Méthode de remplacement des données brutes d'un SMF par les données quantifiées d'une liste d'évènements

Une fois que ces données ont été modifiées, il reste à réencoder ces dernières afin d'exporter un nouveau fichier MIDI quantifié, prêt à être importé et orchestré dans *Sibelius*. Pour finir, afin d'avoir un aperçu plus global de notre outil logiciel, voici sa structure fonctionnelle :

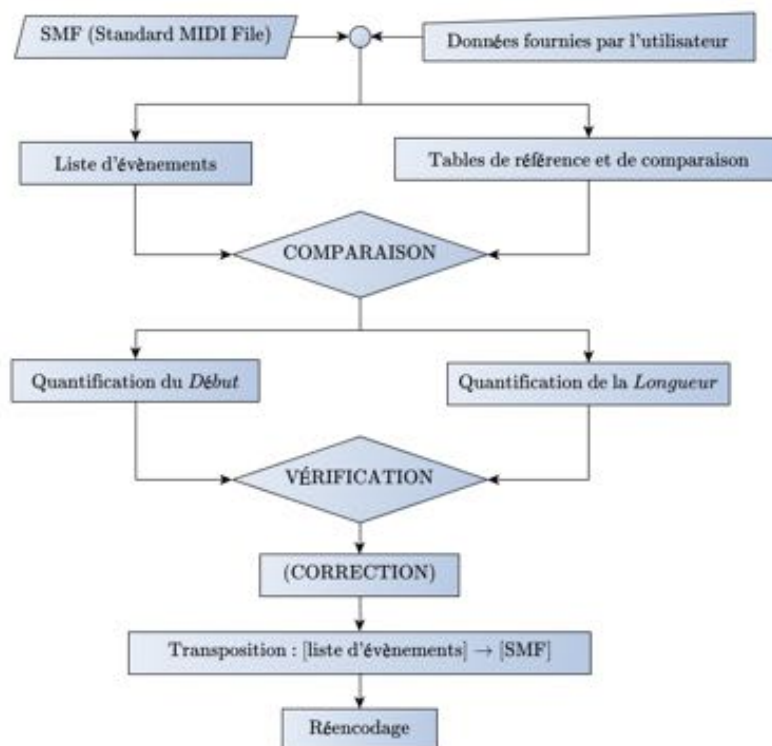


Figure 36 : Structure fonctionnelle de notre outil logiciel

C. Vers une interface transparente : quelques perspectives de développement et d'améliorations

Bien évidemment, le prototype que nous proposons est sommaire, et ne présente en aucun cas une solution complète et définitive aux problèmes de transport des données MIDI issues d'une STAN vers un logiciel de notation musicale. Nous voudrions donc suggérer un certain nombre de pistes de développement et d'améliorations pour un interfaçage entre la STAN et *Sibelius* plus transparent et encore plus rapide. Nous orienterons et présenterons ces perspectives autour de trois axes : la gestion des instruments polyphoniques et des articulations, la préparation des données pour les importer dans une *template*, et enfin la possibilité de procéder à des analyses croisées, et utiliser les corrélations entre plusieurs instruments pour en affiner l'analyse et la quantification.

1. Gestion des instruments polyphoniques et des articulations

Nous nous sommes limités à un seul instrument monophonique, mais pour que notre interface soit complète, il faudra bien évidemment lui permettre de gérer un nombre [presque] infini d'instruments, mais surtout l'élargir aux instruments polyphoniques, qui présentent un certain nombre de problématiques supplémentaires. Parmi celles-ci se trouve bien sûr la possibilité de jouer des accords, qui induit nécessairement la superposition de plusieurs notes. Un instrument polyphonique peut aussi jouer plusieurs voix simultanément : nous pourrions par exemple isoler chaque voix pour faciliter sa notation.

La gestion des articulations est aussi une problématique en elle-même, bien qu'elle ne relève pas de la quantification. Dans le cas des *keyswitches*, il suffirait de renseigner la tessiture de chaque instrument, ou au contraire de renseigner toutes les notes utilisées pour les changements d'articulation et de

mode de jeu, afin de les filtrer et de les supprimer du SMF quantifié destiné à être importé dans *Sibelius*. Par ailleurs, pour un instrument dont les articulations seraient réparties sur plusieurs pistes MIDI, et donc sur plusieurs canaux, nous pourrions imaginer fusionner tous les messages *Note On* et *Note Off* sur le premier canal, afin de n'avoir plus qu'une seule piste pour cet instrument. *A contrario*, il pourrait être envisageable de diviser un instrument polyphonique sur plusieurs pistes, afin d'en faciliter la notation : pour un piano, par exemple, cela reviendrait à diviser la main gauche (registre grave) de la main droite (registre aigu) sur deux canaux par rapport à une note-pivot (le Do_3 , par exemple).

2. Préparation des données pour les importer dans une *template*

Nous avons vu précédemment que l'usage d'une *template* était très répandu dans le flux de travail des compositeurs de musique à l'image. De la même manière, il n'est pas rare qu'un orchestrateur et un copiste utilisent une *template* sur *Sibelius*, afin de ne pas avoir à paramétrer constamment les conducteurs et les parties séparées pour un même projet ou pour une même formation instrumentale. Pour diverses raisons, la *template* du compositeur n'est pas la même que la *template* de l'orchestrateur et du copiste. Plus largement, la *template* d'une STAN n'est pas forcément, voire rarement, identique à celle de *Sibelius*, et vice-versa.

Il pourrait donc être judicieux de réordonner les informations issues de la STAN, afin qu'elles se présentent dans le « bon » ordre, c'est-à-dire dans l'ordre de la *template* de *Sibelius*. Il faudrait aussi transporter les changements de signatures rythmiques, ainsi que les variations de *tempo*. Or, ces dernières sont automatiquement quantifiées à la résolution du SMF au moment de son export depuis la STAN. Cela signifie que les rampes croissantes ou décroissantes, c'est-à-dire les accélérations ou les ralentissements du *tempo*, sont automatiquement quantifiées à la résolution du SMF. En termes de

notation, l'orchestrateur préférera définir des zones d'accélération (« *accelerando* ») ou de ralentissement (« *rallentando* ») plutôt que d'indiquer une nouvelle valeur de *tempo* sur chaque temps. Autrement dit, la gestion d'une variation de *tempo* est laissée à la discrétion du chef d'orchestre : sur le conducteur, on indique un *tempo* initial, une zone de variation, et un *tempo* d'arrivée. Il pourrait donc être pertinent de repérer le début et la fin d'une courbe ou d'une rampe de variation du *tempo*, et ne transporter que ces deux valeurs initiales et finales ; l'idéal serait même de pouvoir inscrire *accelerando* ou *rallentando* sur la zone concernée. Par contre, il ne faut en aucun cas transporter toutes les valeurs de *tempo*, qui devront être supprimées à la main par l'orchestrateur.

3. Analyses croisées et corrélations musicales

Dans l'hypothèse où notre outil logiciel serait capable de gérer de nombreux instruments monophoniques et polyphoniques, il serait dommageable de nous contenter de cloisonner chacun d'eux, ce qui reviendrait à les analyser et à les quantifier uniquement par rapport à eux-mêmes. De nouveaux outils d'analyse permettant de croiser les données relatives à chaque instrument et de vérifier leurs éventuelles corrélations nous permettrait d'affiner la quantification de chacun. Nous pourrions définir des profils d'instruments, par exemple par pupitre, afin de compenser plus précisément les écarts entre des instruments supposés avoir le même rôle musical. L'analyse croisée des données de tous les instruments permettrait aussi de détecter des passages à l'unisson, homo-mélodiques ou homo-rythmiques, et s'assurer alors que tous les instruments soient quantifiés de la même manière dans ces passages.

Toutes ces perspectives d'amélioration et de développement induisent la création de nouveaux algorithmes, plus poussés et plus robustes. Pour conclure, il faut garder à l'esprit que cet outil logiciel, cette interface entre une

STAN et un logiciel de notation musicale doit être une aide et induire un gain de temps pour l'orchestrateur et éventuellement le copiste. Il n'est certainement pas pensé comme une substitution aux tâches artistiques, créatives et techniques de chacun, mais bien comme une assistance, une goutte d'huile destinée à fluidifier une partie du processus de fabrication d'une musique de film.

Conclusion

Comme nous l'avons vu, l'informatique a débuté son implantation dans l'industrie cinématographique et musicale dans les années 1970, pour s'y établir durablement dans les années 1990. Une trentaine d'années plus tard, la mécanique de production d'une musique de film semble bien huilée, et malgré quelques imperfections, les habitudes de travail sont imprégnées et modelées par les nouveaux outils à la disposition des compositeurs. Toutefois, il semble qu'un cycle d'innovations et de chamboulements technologiques soit en train de débiter. Bien entendu, il ne peut en aucun cas être comparé au cinéma parlant ou à la numérisation de l'industrie cinématographique, mais il a tout de même son importance. Ces évolutions en cours ou à venir, sont d'ailleurs symptomatiques des insuffisances ou des défauts que nous avons pointés.

La première innovation qui mérite notre attention est le *Native Kontrol Standard* (NKS), développé par *Native Instruments* depuis 2015. Il s'agit essentiellement d'un « format de *plug-in* approfondi qui permet une connexion transparente entre des instruments ou des effets logiciels et les claviers *Komplete Kontrol* ainsi que le *hardware* [ou « équipement matériel »] *Maschine de Native Instruments* »⁹⁴. En d'autres termes, le NKS est une norme propriétaire qui permet d'implémenter en profondeur un très grand nombre d'instruments et d'effets sur les contrôleurs fabriqués par *Native Instruments*. L'éditeur, qui développe aussi *Kontakt*, ne rompt pour autant pas avec sa politique d'ouverture et de cohabitation, puisque les produits de plus de soixante-quinze développeurs sont aujourd'hui compatibles avec le format NKS. Son but est que l'utilisateur puisse tout contrôler depuis son clavier maître : de l'exploration et du chargement des *patches* aux fonctionnalités et aux paramètres les plus avancés de ses instruments. En plus des commandes qui permettent de contrôler une STAN, tous les boutons du clavier sont

⁹⁴ <https://www.native-instruments.com/fr/specials/komplete/nks/> (dernière consultation le 21/05/2018).

préassignés en fonction du *patch*, de l'instrument ou de l'effet qui est chargé. Indéniablement, le NKS constitue une optimisation du contrôle du studio virtuel par le biais d'un équipement matériel unique et intuitif.

Par ailleurs, nous avons déjà évoqué *Dorico*, le logiciel de notation et de gravure de partitions, développé par *Steinberg* depuis 2016, et qui selon l'éditeur allemand, incarne « l'écriture musicale de l'avenir »⁹⁵. Derrière ce slogan commercial, se cache effectivement une petite révolution, qui prend d'ailleurs le contre-pied de notre outil logiciel d'interfaçage entre la STAN et la partition. En effet, *Dorico* est le seul logiciel de notation musicale à intégrer un séquenceur, qui permet aux utilisateurs de « peaufiner la lecture MIDI en utilisant [notamment] les *Expression Maps* et l'éditeur MIDI »⁹⁶. En d'autres termes, un séquenceur MIDI se situe en sous-couche de la notation musicale, et offre la possibilité de modifier les données MIDI, sans pour autant en modifier la notation sur la partition. Cela implique que l'utilisateur peut légèrement décaler les notes, les superposer, et gérer les différents *patches* de la banque de son utilisée pour jouer la partition grâce aux *Expression Maps*, afin de simuler une performance musicale humaine. En d'autres termes, la programmation MIDI est dissociée de sa notation musicale. *Dorico* introduit également un cloisonnement de l'écriture musicale, de l'édition, et de la gravure de la partition, afin d'optimiser chaque étape et l'interface qui est présentée à l'utilisateur pour chacune d'elle. Toutefois, composer la musique d'un film avec *Dorico* ne tombe pas encore sous le coup de l'évidence, puisque cela impliquerait de ne pas *jouer* à l'image, comme c'est le cas aujourd'hui, mais *d'écrire* à l'image. Par contre, nous pouvons imaginer que *Steinberg* développera à l'avenir une passerelle entre *Cubase* et *Dorico*, afin que ce dernier puisse aisément et naturellement s'intégrer dans le processus de fabrication actuel d'une musique de film.

95 <https://www.steinberg.fr/fr/produits/dorico/start.html> (dernière consultation le 21/05/2018).

96 *ibid.*

Pour conclure, il semble que les développeurs et les éditeurs de logiciels dédiés à la musique [de film] soient en train de s'emparer des problématiques engendrées par certaines vétustés de la norme MIDI. *Native Instruments* et *Steinberg* seront sans doute suivis par d'autres éditeurs à l'avenir, mais quoiqu'il en soit, le développement de la norme NKS et de *Dorico* peut nous laisser présager des mutations et des évolutions à venir. Nous avons tenté, à notre modeste échelle, d'apporter une solution, mais nous pouvons supputer que dans un futur proche, un interfaçage beaucoup plus abouti entre une STAN et un logiciel de notation musicale sera proposé, que la norme MIDI sera modelée ou détournée pour optimiser le transfert de données d'un logiciel à l'autre, ou encore qu'elle sera mise à jour en profondeur, voire abandonnée pour une nouvelle norme. Cette dernière option semble tout de même illusoire, puisqu'il faudrait alors refondre une grande partie des outils qui sont développés et utilisés par toute une profession depuis l'avènement de l'informatique musicale.

Bibliographie

APPLE INC., *Logic Pro X User Guide* [ebook], Cupertino, Apple Inc., 2017.

DAVIS, Richard, *Complete Guide to Film Scoring: The Art and Business of Writing Music for Movies and TV*, Boston, Berklee Press, 1999, 2^{ème} éd. (2010).

FLACH, Paula, *Film Scoring Today: Theory, Practice and Analysis* [en ligne], Mémoire de Maîtrise (sous la dir. de Paul Larsen), Université de Bergen, 2012, URL : <http://bora.uib.no/bitstream/handle/1956/6016/97122214.pdf>.

GILREATH, Paul, *The Guide to MIDI Orchestration*, Burlington, Focal Press, 1996, 4^{ème} éd. (2010).

Groupe de veille et d'analyse « MALRAUX », *Influence culturelle d'Hollywood*, Paris, INHESJ, 2015.

McGUIRE, Sam, *Modern MIDI: Sequencing and Performing Using Traditional and Mobile Tools*, Burlington, Focal Press, 2014.

NOON, Tristan, *From DAW to Score: An Essential Guide to Music in Film and TV, and Transforming Your MIDI Data into a Score Ready for Recording Sessions*, Londres, Tristan Noon Music, 2017.

WIERZICKI, James, *Film Music : A History*, New York, Routledge, 2009.

Sites web

(Dernière consultation le 22/05/2018).

Babylonwaves, <https://www.babylonwaves.com>

MIDI Association, <https://www.midi.org>

Native Instruments, <https://www.native-instruments.com/fr/>

Orchestral Tools, <http://www.orchestraltools.com>

Spectrasonics, <https://www.spectrasonics.net>

Spitfire Audio, <https://www.spitfireaudio.com>

Steinberg (Aide), <https://steinberg.help>

Synthogy, <http://synthogy.com>

Vienna Symphonic Library, <https://www.vsl.co.at/en>

Annexe 1 : Lexique et définition des termes musicaux les plus utiles à la compréhension de notre mémoire

Altération : en musique, elle désigne à la fois le fait de modifier la hauteur d'une note, la notation de cette modification sur la partition, et par extension la « note affectée d'une altération ». Une note non altérée est *bécarre*. Si sa hauteur est abaissée d'un demi-ton, alors elle est *bémol*, et si au contraire elle est augmentée d'un demi-ton, elle est *dièse*.

Anacrouse : on parle aussi de *levée* ; elle désigne une note ou un petit groupe de notes précédant le premier « temps fort » d'une mélodie. Un morceau qui commence « en anacrouse » commencera légèrement avant le premier temps de la première mesure ; c'est par exemple le cas de *La Marseillaise*, ou du premier mouvement de la *Symphonie No. 5* de Gustav Mahler.

Arco : technique utilisée par les instruments à cordes frottées, qui consiste à jouer en frottant les cordes avec l'archet.

Armure : il s'agit de l'ensemble d'altérations réunies à la *clé*. Elle ne peut être composée que de dièses ou que de bémols, et correspond en général à la tonalité principale des mesures qui la suivent.

Bécarre (♮) : il s'agit d'une *altération* qui indique que la note à laquelle elle est assigné doit être jouée à sa hauteur naturelle.

Bémol (♭) : cette *altération* implique un abaissement d'un demi-ton chromatique de la hauteur naturelle de la note qui lui est associée.

Clef : placée au début de chaque portée, elle indique la hauteur des notes associées à chaque ligne. La figure ci-dessous indique la position du Do_3 dans huit clefs différentes. On remarquera que les clefs présentent trois figures différentes, avec de gauche à droite la *clé de fa*, la *clé d'ut* et la *clé de sol*.



Col legno : technique de jeu réservée aux instruments à cordes frottées, qui est exécutée en attaquant les cordes avec le bois plutôt qu'avec les crins de l'archet.

Glissando : terme d'origine italienne qui désigne soit un glissement continu d'une note à une autre, soit le passage d'une note à l'autre par un groupe de notes intermédiaires.

Legato : mode de jeu qui consiste à lier les notes successives, afin qu'il n'y ait pas de silence entre elles.

Marcato : mode de jeu où les notes sont martelées.

Nuance : il s'agit d'un signe noté sur la partition qui indique l'intensité sonore d'une note, d'une phrase, ou encore d'un passage entier d'une œuvre musicale. Ainsi, les nuances permettent à l'interprète de restituer la dynamique de l'œuvre telle qu'elle a été imaginée par le compositeur.

Pizzicato : technique utilisée par les instruments à cordes frottées qui consiste à pincer les cordes avec les doigts au lieu d'utiliser l'archet.

Rubato : ce terme italien, qui signifie « dérobé », commande d'abandonner la rigueur de la mesure selon l'inspiration de l'interprète ou du chef d'orchestre ; ce geste musical est caractéristique du style romantique.

Signature rythmique : appelée aussi *chiffage de la mesure*, elle indique le type et la durée des temps dans les mesures qui la suivent, jusqu'au prochain changement de signature rythmique. Les deux nombres du chiffage forment une fraction, dont l'unité de valeur est toujours la ronde.

Spiccato : technique réservée aux instruments à cordes frottées qui consiste à faire rebondir l'archet légèrement sur les cordes.

Staccato : ce terme signifie « piqué », et désigne un phrasé en notes détachées, ce qui signifie qu'il y a une suspension entre chaque note. C'est le contraire du *legato*.

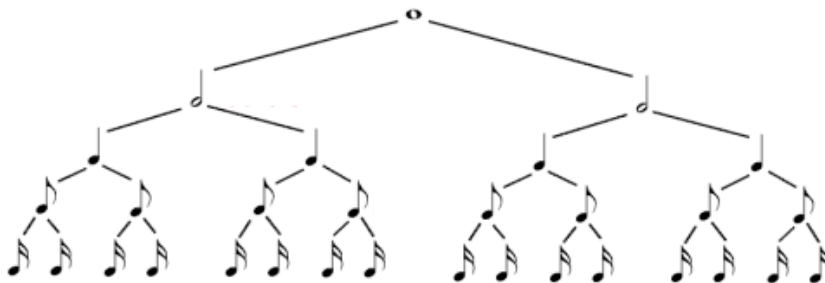
Staccatissimo : mode de jeu détaché vif, encore plus marqué que le *staccato*.

Tempo : vitesse d'exécution d'une œuvre, qui correspond à la vitesse de la *pulsation*, c'est-à-dire au battement d'un temps musical régulier, qui garantit l'égalité entre les temps.

Tenuto : consiste à tenir une note dans le temps.

Tutti : dans le cas d'un orchestre, il définit un passage où tous les instruments jouent ensemble.

Valeurs rythmiques : l'ensemble des valeurs rythmiques utilisées dans le langage musical occidental (ronde, blanche, noire, croche, etc.) sont définies les unes par rapport aux autres. L'arbre ci-dessous représente les premières subdivisions des valeurs rythmiques simples : une ronde se divise en deux blanches, la blanche en deux noires, la noire en deux croches, etc.



Annexe 2 : Quelques exemples de nombres entiers codés en VLQ donnés par la spécification du SMF

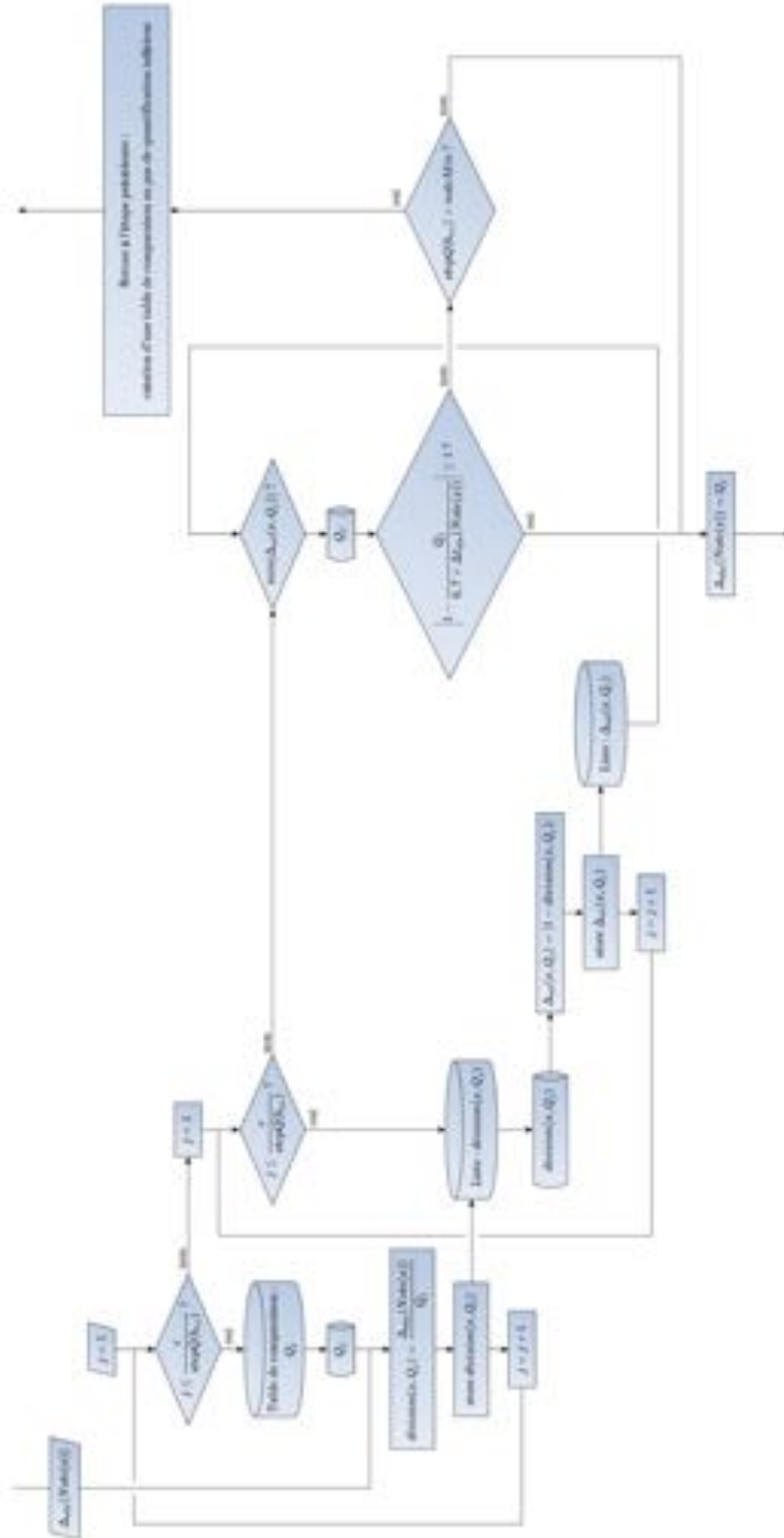
Entier	VLQ
0x00000000	0x00
0x0000007F	0x7F
0x00000080	0x81 0x00
0x00002000	0xC0 0x00
0x00003FFF	0xFF 0x7F
0x00004000	0x81 0x80 0x00
0x001FFFFFF	0xFF 0xFF 0x7F
0x00200000	0x81 0x80 0x80 0x00
0x08000000	0xC0 0x80 0x80 0x00
0xFFFFFFFF	0xFF 0xFF 0xFF 0xFF

Annexe 3 : Liste des contrôles continus (CC) les plus utilisés dans le cadre de la composition de musique à l'image

Valeur	Type	Description ⁹⁷
1 (0x01)	Modulation	Roue de modulation : elle est souvent utilisée pour doser le taux d'effet de <i>vibrato</i> ou de <i>tremolo</i> d'un son, ou encore pour régler sa brillance.
2 (0x02)	Breath Controller	Contrôleur de souffle : il s'agit d'un capteur placé dans le bouche du musicien qui est sensible aux variations de pression par le souffle, utilisé notamment pour moduler les sons de cuivre.
7 (0x07)	Main Volume	Volume : positionne le volume général du canal sélectionné.
10 (0x0A)	Pan	Panoramique : positionne le son du canal sélectionné dans l'espace panoramique stéréophonique.
11 (0x0B)	Expression Controller	Expression : positionne le volume relatif du canal sélectionné en rapport avec le niveau de volume général ; elle travaille en pourcentage du volume courant. L' <i>Expression</i> est donc plutôt utilisée pour régler des <i>crescendi</i> / <i>decrescendi</i> alors que le CC#7 est utilisé pour régler le volume « maximal » d'un canal. Quand l' <i>Expression</i> est égale à 100 %, le volume réel correspond au niveau du réglage du volume général (CC#7) du canal, tandis qu'à 0 %, le son est coupé.
64 (0x40)	Damper Pedal (sustain)	Pédale de maintien (<i>on/off</i>) : provoque le maintien des notes du canal sélectionné, lesquelles vont être relâchées par le musicien, de la même manière qu'une pédale de piano permet de conserver la résonance du son comme si les touches n'avaient pas été relâchées.

⁹⁷ D'après le dossier *Le MIDI : les MIDI control changes* publié sur Audiofanzine le 23/08/2002 : <https://fr.audiofanzine.com/mao/editorial/dossiers/le-midi-les-midi-control-change.html> (dernière consultation le 20/05/2018).

Annexe 4 : Algorithme de quantification du Δt_{abs} d'une $Note(x)$ donnée



Annexe 5 : Algorithme de quantification de la longueur d'une *Note(x)* donnée

