

## コンテンツクリエイターのための 著作権フリー音楽クリップ生成システム"FMC<sup>3</sup>"

### "FMC<sup>3</sup>" : Copyright-free Automatic Composition System for Contents Creators

長嶋洋一

Yoichi NAGASHIMA

ASL/SUAC

nagasm@computer.org

**概要** FLASHやショートムービーなどのコンテンツを制作するクリエイター/デザイナーが、著作権について心配することなく音楽トラックの素材として自由に利用できる、「コンテンツクリエイター共同体のためのフリー音楽クリップ」生成システムFMC<sup>3</sup>(Free Music Clip for Creative Common)を開発した。多数のFLASHコンテンツの調査から求められる音楽クリップの特徴を検討抽出し、本質的に著作権問題を起こさない、従来の自動作曲システムとは異なるアプローチによるアルゴリズム作曲の手法によって、音楽的専門知識の必要なくコンテンツ制作に適した音楽トラックを手元で自動生成/アレンジするシステムをMacintosh/Windows両環境で提供した。生成される音楽クリップのパラメータを100バイトのplain textに整理圧縮して、検索/交換に役立つ体系を実現した。

**Abstract** : I developed "Automatic composition system of the music track that the creator was able to use for FLASH contents freely as the copyright did not worry". I have examining extracted the feature of the requested music clip from the investigation of a lot of FLASH contents. Next, I developed the system that generated the music track automatically by the technique of an algorithmic composition different from a past automatic composition system in both environments of Macintosh/Windows. The creator generates a lot of music clips and can arrange it freely without requesting musical expertise.

**キーワード** : 自動作曲 音楽クリップ コンテンツクリエイター アルゴリズム作曲 著作権

**Keywords** : Automatic Composition Copyright-free Algorithmic Composition Max/MSP

#### 1. はじめに

音楽情報科学研究における古典的テーマ「自動作曲」、さらにエンタテインメントコンピューティングやメディアアート研究に関連して、「誰でも手軽に作品系FLASHコンテンツの音楽部分を自動生成するシステム」を開発した。マルチメディア作品系コンテンツを「自分で作る」というエンタテインメントの時代に課題となる音楽パートの創作を、著作権の問題をクリアし、音楽的な専門知識を要求することなく、容易にコンテンツに適した「使える」音楽を自動生成するシステムの実現を目指した。

##### 1.1 研究の背景(1) - 自動作曲の歴史

コンピュータによる自動作曲の研究は1959年頃に始まり[1]、歴史はほぼ半世紀になる[2]-[16]。初期は和声分析や統計確率やデータベースの手法から「誰々っぽい作品の自動作曲」が指向され、続いてマルコフ連鎖・ニューラルネット・遺伝アルゴリズムなどの学習システム的な手法が検討された。また音楽形式分析やフラクタルや感性情報処理的アプローチ、知識情報処理、音楽認知や音

楽学との協力も続いている。ここ10年ほどは、音楽情報処理の力点がインターネット時代の要請から楽曲データベースの検索や音響符号化/伝送などに移り、自動作曲という永遠のテーマは「現代音楽」とともに、やや停滞した印象もある。

筆者はComputer Musicの作曲家/研究者として、主としてメディアアート領域での作品創作発表/研究を行ってきた[17]。研究テーマの多くはインタラクティブやシステム開発で、自動作曲研究への直球勝負は敢えて避け、一方でネットワーク利用の音楽セッション[18][19]、コンテンツにおける映像と音楽のビートの同期[20][21]など、コンテンツ関連領域も研究テーマとしてきた。

##### 1.2 研究の背景(2) - FLASHの時代

創作や研究においてマルチメディア系コンテンツの開発環境を調査する中で、筆者はFLASHに注目した。FLASHコンテンツの多彩な表現と軽さはパソコンだけでなく、携帯電話やインターネット対応の電子機器への組み込みにも好適である。多くの企業のWebサイトにFLASHが活用されるようになり、2年後には到来するであろう「クリエイション(作品創造)の大衆化」の時代には、「デジカメ写

真を自動スライドショー化」「ホームビデオを半自動編集してmovie作品化」「FLASH作品のお手軽制作ツール」など、一般大衆がマルチメディア作品系コンテンツを自分で作る、という新しいエンタテインメントが出現すると期待される。

### 1.3 従来の「音楽データが使えない」理由

ここでの焦点は、「使える音楽データをお手軽に欲しい」という要請である。DTM (打込み音楽) ホビーストや音楽の専門知識があれば、MIDI機器やDTMソフトによって、オリジナルの音楽を作って楽しむ環境は、ITの進展とともに充実してきた。しかし音楽の専門家でない一般大衆・ビジュアルデザイナー・クリエイターの希望は、現在のところ、以下の3点において解決されていない。

#### 1.3.1 入手する - 著作権のため使えない

いかにお気に入りの楽曲であっても、プロアマ問わず他人の作った音楽は、作品を公開する用途に無断では使えない。自分で「耳コピー」したり市販の楽譜を打ち込みしてMIDIデータ化しても駄目である。クリエイターの卵が行っている対応は、(a)知り合いに作曲・提供してもらう、(b)著作権フリー音楽集の利用、(c)音楽シェアウェア(有料で使用許諾)の利用、などである。その結果「あの作品とまた音楽が同じ」等のFLASH作品があちこちに生まれ、やや淋しい創作となってしまう。

#### 1.3.2 市販ソフト - 3つの問題点

市販の「自動作曲/作曲支援」製品にも、多数の音楽演奏「断片」データベースから組み合わせで切り貼りし、音符を打ち込むことなくお手軽に作曲する、という音楽生成ソフトウェアは存在する。しかし素材がメーカー/作曲家/編曲家の提供する既存の楽曲データでは、著作権の気掛かりから逃れられない。これが第一の問題点である。

MIDIトラックとリアル音源トラックの混在するシーケンサに、多数の著作権フリー・サウンドフレーズを選択して並べて繋げる、というMac標準の"GarageBand"はシンプルで画期的な製品である。しかし多くのユーザが好むフレーズはライブラリのごく一部に集中する事が多く、パリエーション不足のためにオプションフレーズ集を別途購入するという新たなコスト負担が必要となる。これが第二の問題点である。

"GarageBand"のリアル音源のようなサンプリング方式の場合、最近ではピッチ変更・時間的移動などの編集機能が強化されているが、本質的な困難が残っている。MIDIであれば全てミックスされた演奏情報の一部だけをルパート(テンポの局所的な変化)させる事は容易だが、サウンドファイルに混じった中から特定のパートだけを抽出分離して時間的にルパートさせるのは、音楽情報科学における巨大なテーマ「音源分離」そのもので、使えるレベルで実用になっていない。サウンドファイル全体をピッチを変えずに部分的に時間伸縮するのは比較的容易だが、音楽のノリ・タメ等の表現は特定のパートだけが時間的に他パートからズレ

ることで生まれるので、この課題はリアル音源ベースのソフトウェア(市販/研究)でいまだ解決されていない。これが第三の問題点である。

#### 1.3.3 研究者のシステム - 面白くない

複数の素材(著作物)を用意し選択し接続する、という手法には著作権の負い目がある。そこで自動作曲システムには、音楽演奏情報(素材)を持たないゼロからスタートして新たに音楽演奏情報を生成する、というアルゴリズム作曲[11]の手法が本命であろう。このテーマの研究は、国内では現在でも多くの大学等で行われているが、工学系の研究者がITの応用例としてたまたま音楽を使ったような報告には批判も多い。その多くは、1世紀前から既知の音楽理論/経験則すら十分に調査せず、音楽心理学の領域で基礎/常識とされるルール(関連学会等に参加すれば容易に検討可能)を考慮していない。その結果、一般的な聴取者にとって不自然な「自動作曲」研究(実質的な内容は1970年代から進歩なし)が繰り返されている。音楽を分かっていない工学者の開発する自動作曲システムには、今後もあまり期待できない。

## 2. 調査とシステムの基本戦略

### 2.1 作品系FLASHの収集と分類

本研究は「あらゆる音楽を自動作曲」などという無謀な野望は抱いていない。市販の高機能シーケンソフトやマニアックな自動作曲ソフトは、「どのような音楽スタイルでも作曲できる」万能性・網羅性を標榜することで、専門的でないユーザの希望から乖離してきた。そこでまず対象を「作品系FLASH」と明確に限定し、企業/商用サイトに多い「WebサイトのナビゲーションFLASH」を対象から除外して調査を開始した。

基本的にはそれ以外のコンテンツは一応FLASHの「作品」である。ただし本研究では「どっきり系/エロサイト系/グロサイト系/ゲーム系/外国語能力が必要」などのFLASHも除外した(理由・詳細については文献[22]を参照)。調査では約5ヶ月かけてインターネット上から約7500本のFLASHを収集、対象外FLASH等を捨てた約4500作品を対象として分類/検討した。この大部分は、Wikipediaの分類で言えば「アニメーション」「ミュージックビデオ」に属する(詳細は文献[23])。

### 2.2 作品系FLASHの音楽の特徴検討

収集した多数の作品系FLASHから約1500本ほどに精選し、実際にメディアコンテンツの制作を行っている学生を被験者とした評価実験からその特徴の抽出を試みた。具体的には、静岡文化芸術大学デザイン学部の学生/大学院生9名に対して、多数のFLASHを視聴し、面白い・イケてる等positiveな印象、つまらない等negativeな印象の作品のピックアップ(推薦)、視点として音楽に注目した場合どんな部分がアピールしたのか、を自由形式でアンケート回答することを求めた。この作業により、音楽の専門家でないデザイナーに想定する音楽

知識のレベルもあわせて取材した。

これを受けて、本研究で開発する作品系コンテンツの特徴に対応した、音楽クリップの自動生成(作曲)の指針を検討した[24]-[26]。作品系コンテンツ(音楽付モーショングラフィック作品)の制作方法には、(1)実写やCGの映像トラックを制作した後に対応したサウンド素材を切り貼りしていく[絵先]、(2)サウンドトラックに楽曲を置いてから対応した各種映像素材を切り貼りしていく[音先]、という大きく2種類のアプローチがある。本システムはMotionDiveやLifeWithPhotocinemaと同じく、先にBGMパートが完成して与えられる後者[音先]のアプローチを対象としている。この場合、絵コンテに相当する作品全体の概要構想に対応して

- ・全体/部分の時間的長さ(尺)
- ・イントロ、エンディング
- ・リピート、ループ、ブレイク
- ・ブリッジ、フェード

などの音楽的要素・音楽構造があらかじめ先に決められる、という独特の傾向と要請がある事が判った。例えば「30秒のCM」という規定は「尺」として最初に決定され、これに応じたBGMを生成してサウンドトラックに張り込み、これに合わせて映像トラックを編集制作する。

本システムのバージョン1ではごく一部のみの実装となったが、設計段階で基本的な可能性として考慮した要素として、

- ・音楽パートのある部分だけを時間的に伸縮
- ・映像と音楽のビートの微調整
- ・「ノリ」「タメ」「グループ」

などの表現可能性を考慮したシステム構造を指向した。これは、既に20年の歴史があるMIDIがもっとも得意とする表現の領域であり、「GarageBand」のリアル音源パートのようなサウンドファイル切り貼り方式(多数のパートのサウンドが混じった状態から特定のパートを切り出して編集する必要がある)では困難である。

## 2.3 システム開発研究の基本戦略

### 2.3.1 自動生成アルゴリズム

システム開発の基本方針として、音楽演奏情報データベースの切り貼りだけでなく、乱数をベースとした自動生成アルゴリズムを採用し、本質的に既存の楽曲情報を一切利用しないこととした。ただし、でたらめな乱数そのままを音楽に置換するのではなく、多様な音楽ヒューリスティクスを随所に反映させており、単純に乱数を用いただけのシステムとは基本的な立場が異なっている。これと同時に、生成された楽曲の特徴情報を圧縮して生成データの中に埋め込み、容易にインターネット検索できる手法も開発した。この手法により、生成パラメータ(非常に小さい)をメール等で交換する、という新しい利用法も提案できた。

### 2.3.2 MIDIベースの自動作曲

音楽データのアルゴリズム作曲自動生成をMIDI情報ベースで行うこととした。バージョン1ではごく一部にしか実装されていないものの、これによ

り、特定パートの演奏を時間的に移動させる「ノリ」「タメ」などの表現への実現可能性を持つ。出力MIDI情報(標準MIDIファイルSMF)からは、ユーザが好みの音源システムで音響データに変換し、さらにFLASH化のためMP3データ化へと活用できることとした。

### 2.3.3 「主役でない」音楽

対象を「作品系マルチメディアコンテンツの音楽パート」と限定した自動作曲コンテンツにおいて、本質的に音楽パートは唯一の主役ではない。モーショングラフィック、モーシオンタイポグラフィ、等のビジュアルあるいはダンサブルな主役のメディアとともに、コンテンツの聴取者に快適な音楽を提供して躍動感/快適感/没入感などの演出効果を目指す。そこで、主役としての音楽であれば骨格部分に据える筈の「メロディー/和声」をスタートラインにしない方針をとった。快適なBGMに必要なのはまずドラム系のリズム、そして「活躍するベース」(詳細は後述)である。作曲を和声やメロディーから始める、という従来の自動作曲に対して、本システムではここが最大の相違点である。

### 2.3.4 著作権について

楽曲であればその誕生とともに音楽著作権が生まれる、というのは、著作権法により著作者が人間の場合に限る[27]。コンピュータのアルゴリズムによりいくらかでも自動生成される多数のミュージッククリップには、本質的に著作権が存在しない。そこで基本の方針として「フリー音楽クリップ」と位置付けた。

## 2.4 開発環境と対象プラットフォーム

本システムのプラットフォームは、Mac OSXおよびWindowsXPである。システムの基幹部分は両環境に共通したMax/MSPで開発し、関連したツール類はXcodeなどのツール(C++)で開発して連携させた。

公開サイト[28]でダウンロードできるモジュールはアーカイブ形式(\*.SIT/\*\*.ZIP/\*\*.LZH)であり、これを解凍したフォルダがそのまま実行環境となる。フォルダの中にある"FMC3\*\*\*"という実行プログラム(\*.app/\*.exe)をダブルクリックするとソフトウェアが起動する(図1)。

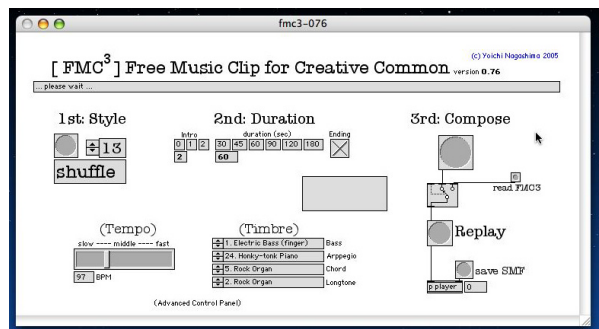


図1 本システムの起動画面例

内部MIDIバスを持つMac OSX版は、以上である。WindowsXPにはOSにこの機能がないため、さらに以下のような対応が必要となる。(1)USB-MIDIインターフェイスを用意し対応USBドライバをインストールする。(2)MIDIインターフェイスをパソコンに接続し、その"MIDI IN"と"MIDI OUT"とをMIDIケーブルで繋ぐ。以上がMac OSX版との違いで、他のセッティングや機能は全て同等である。なお、公開サイト[28]ではソースプログラムや詳細な解説も全て公開しており、Max/MSPを持つ人はプログラム自体を研究/改造することも可能とした。

## 2.5 ネーミング「FMC<sup>3</sup>」

本システムを「FMC<sup>3</sup>」(Free Music Clip for Creative Common: クリエイターのためのフリー音楽素材、発音「エフ・エム・シーキューブ」と名付けた。本システムが生成するMP3(音響)、あるいはこれに対応するSMFデータは、筆者の個人的な判断としてはあくまで音楽素材であって、音楽作品(楽曲)ではない。音楽著作権の存在しない"Music Clip"であって、人間が作曲した楽曲"Musical Piece"ではない。つまりFMC<sup>3</sup>の利用・改編・配布は全てフリーというのが基本姿勢で、利用したい作家に提供している。なお"Creative Commons" [29]とは基本的に無関係である。筆者は"Creative Commons"の活動についてはおおいに共感・協賛できるが、本研究については、上記の判断から正式にCCの枠組みに乗ることは敢えて避けた。FMC<sup>3</sup>の直訳は「創造的な共同体のためのフリー音楽クリップ」である。

## 3. FMC<sup>3</sup>の使用方法

ここでは、FMC<sup>3</sup>の使い方からその概要を説明する。なお、最初にだけ設定する必要のある「MIDI環境の設定・OverDriveの設定」については、FMC<sup>3</sup>公開サイト[28]を参照されたい。

### 3.1 お手軽に「お任せ」自動作曲

#### 3.1.1 ステップ(1) - スタイルを指定

まず最初に、図1の画面に"1st : Style"とあるボタンをクリックすると、36通りのいずれかのスタイルパラメータがランダムに選ばれる。表示される数字(0-35)は、大きいほどベースのフレーズが複雑な音候補から選ばれる。FLASH評価実験の検討から、3種類の「スタイル」は音楽的専門知識がなくても一般ユーザが感覚的に選べるために、自動作曲の起点とした。対応した「テンポ」「音色」その他のパラメータを具体的に設定するのは専門的なので、まず初期値としてシステムが妥当な範囲の値にランダム設定する。

#### 3.1.2 ステップ(2) - 「尺」を指定

次に、図1の画面に"2nd : Duration"とあるブロックを設定する。"Intro"は、「無し/1小節/2小節」のイントロを選ぶ。"duration"とは音楽ク

リップ本体部分の「尺」(長さ)で、30秒から180秒までの6種類のいずれかとして指定する。"Ending"のボタンに"x"印があれば(クリックON/OFF)、最後に「お任せ」のエンディングが付く。

前述のFLASH評価実験の検討から、本システムが対象とするコンテンツでは、サウンドトラックを決めた後に対応した各種映像素材を切り貼りする[音先]のアプローチなので、コンテに相当する作品全体の構想に対応して「全体の長さ(尺)」を最初に指定する、としたからである。

#### 3.1.3 ステップ(3) - 自動作曲開始!

最後に、図1の画面に"3rd : Compose"とあるボタンをクリックする。これで、システムは他の全てのFMC<sup>3</sup>パラメータをランダムに生成し、さらに引き続き、その結果を試聴(パソコンの内部ソフトウェア音源によるリアルタイム生成、同時にリアルタイム記録)できる。画面には刻々とコードネームが表示され、スペースキーでストップできる。

## 3.2 微調整とリプレイ

音楽クリップの「テンポ」とそれぞれのパートの「音色」は、演奏していない時に、図1のメイン画面にあるスライダとボタンで自由に変更することができる。この変更結果を試聴する場合には、画面の"Replay"ボタンをクリックして、気に入るまで何度でも変更(アレンジ)して試せる。

図1のメイン内の"Advanced Control Panel"をクリックするかショートカットの「a」により、図2のような"Advanced Mode"画面が加わり、さらに以下のようないろいろなニュアンスを変更設定することができる[28]。

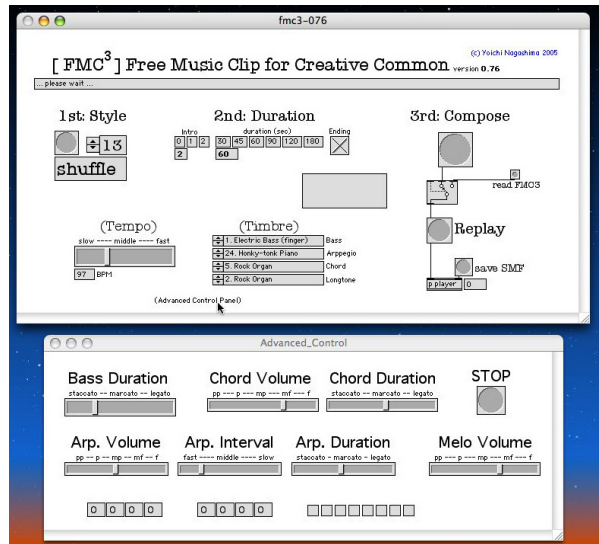


図2 Advanced Modeの加わった画面

#### 3.2.1 ベースの"Duration"

Durationとは、FMC<sup>3</sup>では「音楽クリップそのものの長さ(尺)」としても使うが、ここではMIDIの概念で言う「ノートONからノートOFFまでの時間」



つまり「1音ごとの長さ」である。ベースパートのDurationをいろいろに変えることで、「ノリ」など音楽的表現の領域である、スタカート/マルカート/レガートのアレンジも可能となる。FLASH評価実験の検討から、音楽的専門知識(コード理論など)の無いグラフィック系のデザイナーでも、耳で聞いて感覚的にこの表現の違いを理解し、コントロールできるものとして採用した。

### 3.2.2 コードパートの表現

FMC<sup>3</sup>のコードパートに対して、“Chord Volume”と“Chord Duration”の2つがある。固定のドラムとベースの音量に対して、“Chord Volume”により相対的なコードパートの音量が変更できる。またコードカッティングパートのDurationを変えることで、音楽的表現としてスタカートからレガートまでのアレンジも可能である。

### 3.2.3 アルペジオパートの表現

FMC<sup>3</sup>のアルペジオパートのアレンジ項目として“Arp. Volume”, “Arp. Interval”, “Arp. Duration”の3つがある。VolumeとDurationはコードパートと同様であるが、アルペジオパートだけにある“Arp. Interval”とは、アルペジオの1音1音の時間的な間隔を意味する。これは音符の長さ(テンポとともに変化)に関係なく絶対的な時間として設定され、人間の手で演奏したような「遅れ」のニュアンスや、音楽の「ノリ」「タメ」に相当する時間的なずれを実現できる。

### 3.2.4 ロングノートパート

FMC<sup>3</sup>のロングノートパートには、メイン画面の「音色」の他に“Advanced Control Panel”に“Melo Volume”というスライダーがあり、固定のドラムとベースに対する相対的な音量が変更できる。

### 3.3 MIDIファイルとして保存

生成・試聴してみたFMC<sup>3</sup>が気に入った場合には、メインスクリーンの右下にある“save SMF”というボタンをクリックして保存できる。「保存するファイル名と場所」を聞いてくるので、Formatを「MIDI file」に、ファイル拡張子を「\*.mid」にして保存する。保存されたMIDIファイルをダブルクリックすれば、QuickTimeプレーヤ等ですぐに再生(音質良好)できる。

## 4. FMC<sup>3</sup>の生成アルゴリズムの詳細

### 4.1 FMC<sup>3</sup>パラメータ

FMC<sup>3</sup>の自動作曲・音楽クリップ生成パラメータは「1行メイルとして送れるサイズ」を念頭に、全体で「100文字」と基本設計した。この1文字にエンコードされる情報は、「半角数字の0から9まで」「半角大文字のAからZまで」という、合計36通りとして割り当てられる(36進数)ので、生成さ

れる音楽クリップごとにパラメータのいずれかが異なる場合、理論上「36の100乗(6の200乗)」の組み合わせがある。実際には一部を管理情報(version/revision等)に使用したり、開発の余地としてReservedとする領域もあり、当面は「36の80-90乗」程度とするが、それでも十分なバリエーションを提供できると考えている。なお、FMC<sup>3</sup>公開サイト[28]において、全パラメータの一覧表と個々の定義を詳細に解説している。

### 4.2 FMC<sup>3</sup>生成の基本的フロー

「FMC<sup>3</sup>生成ソフトウェア」そのものここでは単にFMC<sup>3</sup>と呼び、音楽クリップ自動生成のアルゴリズムを詳細に解説する。図3はその生成フローである。

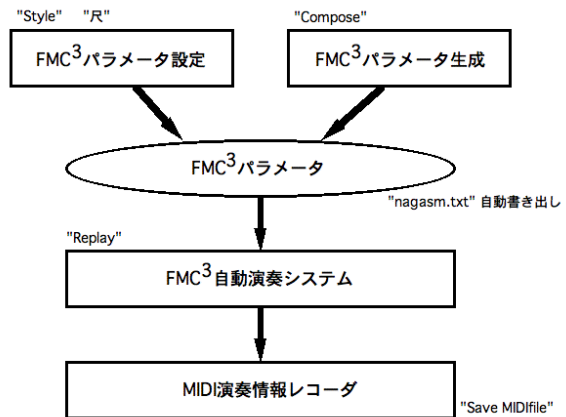


図3 本システムの音楽クリップ生成フロー

「100文字の36進数」であるFMC<sup>3</sup>パラメータには大きく2種類、「設定・変更できる」ものと「自動作曲においてランダムに生成される(お任せ)」ものがある[28]。「尺・テンポ・音色・BassのDuration」などは「設定・変更できる」もので、気に入るように変更してリプレイできる。一方「ドラムパターン・ベースフレーズ・コード進行・イントロ/エンディング」などは、“Compose”コマンドのたびにシステムが乱数をもとに自動生成し、個別の修正変更はできない。この「お任せ」という仕様はFMC<sup>3</sup>の基本ポリシーである。

FMC<sup>3</sup>パラメータは先頭に“NAGASM2005”という10文字の加わったplain textファイル“nagasm.txt”(予約名)として常に自動保存される。これに続いて(あるいは“Replay”コマンドを受けて)、システムはこのFMC<sup>3</sup>パラメータに基づいて、リアルタイム自動作曲演奏(コードネームの表示付き)を行う。コンピュータ内蔵音源を用いたリアルタイム処理の関係でサウンドのクオリティはあまり高くないが、FMC<sup>3</sup>の概形を確認できる。

このFMC<sup>3</sup>リアルタイム自動作曲演奏は、同時に刻々とMIDI演奏情報レコーダに記録される。演奏

が終了した場合、あるいは強制ストップしたところで、"Save SMF"コマンドにより演奏情報を「標準MIDIファイル」形式(format 0)で保存できる。MIDIファイルはQuickTime Playerなどソフトウェア音源システムにより、より良好な音質のサウンドとして再生でき、AIFF/WAV/MP3など必要なサウンドファイルに保存できる。この音質の違いの理由は、QuickTimeなどのソフトウェア音源システムは、内部的には再生に先立ってMIDIデータを先読みしてサウンドに展開(エフェクト・イコライジングなどを含む)し、時間をかけて良質の音響データに変換する余裕があるためである。

#### 4.3 3種類の「スタイル」とテンポ

FMC<sup>3</sup>パラメータ(m=7)により、もっとも基本となる3種類の「スタイル」を決定する。0-35の範囲のパラメータ(m=7)を3で割った剰余nにより、

- ・8beat (1小節を8分音符8個に分割) n=2
- ・shuffle (1小節を3連符12個に分割) n=1
- ・16beat (1小節を16分音符16個に分割) n=0

の基本スタイルが、生成された音楽クリップごとに固有のテンポですと進行する。

同時に、このパラメータ(m=7)を3で割った商(0-11)が大きいほど、ベースパターンを構成するノートの候補が多種になる。ベースラインの音候補とは、もっとも単純なものから

- ・コードのルート音(オクターブ違い)だけ
- ・ルートと5度音
- ・多くのテンションノーツを含むスケール
- ・全音音階Whole Tone Scale

などで構成されていて、パラメータ(m=7)の値が大きいほど、ベースのフレーズにテンションや不協和音が含まれる確率が高くなる。

この3種類のスタイルごとに、4/4のメトロノーム速度BPMを、50から200までの範囲に分布する36段階の値としてテンポを設定する。xをパラメータ(m=10)の値、yを対応BPMとすると、テンポ範囲を指数関数的に等分割した(x,y)の組み合わせは、

$$(x,y) = (0,50), (1,54), \dots, (33,192), (34,196), (35,200)$$

などとなっている[28]。

#### 4.4 「尺」とIntro/Break/Ending

FMC<sup>3</sup>のパラメータ(m=9)は、音楽クリップ全体の基本となる「尺」と、イントロ/エンディングの有無を指定する。「尺」は6種類あり、パラメータ(m=9)を6で割った商(0-5)に応じて順に「30, 45, 60, 90, 120, 180(秒)」という長さである。実際には、テンポに応じてFMC<sup>3</sup>の本体部分(イントロは除外)の長さが2小節単位としてこの時間を越えたところで「終了」(Ending = ONならエンディング)に進むので、厳密に音楽クリップ全体の長さを規定するものではない。

イントロとエンディングについては、FLASH評価実験の検討結果から、対象とするコンテンツにおいてフェードイン/アウトする事も多く重要度は低

いと判断して、個別に作成せずに「36種類のパターンの中から選ぶ」のみとした。

イントロは、0-35の範囲のパラメータ(m=9)を3で割った剰余nにより「イントロ無し/1小節イントロ/2小節イントロ」となる。3種類のスタイルごとに、自動作曲ステップで乱数により生成されるパラメータ(m=75)によって36種類のドラムパターンが選択され、「1小節イントロ」そのもの、あるいは「2小節イントロの2小節目」として使用される。2小節イントロの1小節目は、全てのスタイルで「4分音符のリムショット4発」とした。

エンディングは、パラメータ(m=9)を6で割った剰余が3以上であればON、2以下であればOFFとして組み込んだ。このエンディングは3種類のスタイルに共通で、このパラメータのみ、ランダム生成でなく自動作曲のたびにインクリメント(+1)される。36種類のうち24種類を将来の拡張用に予約としたため、バージョン1では12種類ごとに同じタイプが繰り返される。演奏が「尺」で規定した時間を越えて2小節単位の演奏が終わると、最後に鳴っていたコードから完全5度下行のドミナントモーションで進むrootのロングトーンが鳴り、アルペジオパートの音色によるスケールの繰り返しでdecresc.することで「エンディング」とした。

FMC<sup>3</sup>ではさらに、FLASH作品によくある演出パターンのために、「尺」の時間をおよそ黄金比に分割する地点に「ブレイク」として、イントロパターンのドラムソロ(尺の長さに応じた回数)を挿入する仕様とした。全体のDurationを黄金比に分割する地点を音楽的变化点にするというのは、バルトークなど多くの作曲家が愛好した古典的なアイデアである。FMC<sup>3</sup>バージョン1ではパラメータ(m=74)の値を乱数により生成し、1/36の確率でゼロの時だけブレイク無しとして、それ以外はイントロのドラムソロと同じパターンとなる。

#### 4.5 各パートの音色設定

FMC<sup>3</sup>のパラメータ(m=12)からパラメータ(m=15)までの4つに対して、各パートごとの楽器音色を設定し、この楽器音色名を画面内に表示した。アップダウンボタンを操作することで、各パートの音色を設定変更して自動作曲ステップに進んだりReplayしたりできる。パートごとにパラメータの値に対応した具体的な音色名(楽器名)の一覧は公開サイト[28]を参照されたい。

標準的なMIDI音源はパートごとに128種類の音色を設定できるが、本システムではパラメータ圧縮のためにパートごとに128音色の中から36種類を選択して提供した。この音色に不満な場合、生成したMIDIファイルを一般のMIDIシーケンサに取り込んで、自在の音色選択、さらには異なった音色バンクに交換できる。この詳細についても公開サイト[28]で詳しく解説した。

#### 4.6 ドラムパートの自動生成

FMC<sup>3</sup>ではすべての音楽クリップは4小節単位で進

行し、コードが小節単位で移動(和声進行+転調)しつつ、ベースのフレーズ生成は2小節単位で同じパターンを繰り返し、ドラムは1小節単位で同じパターンを繰り返す。これは多数のFLASHコンテンツの分析から得られた最大公約数的な仕様である。

ベースとドラムのリズムの分解能は、「8ビート/16ビート/シャッフル」の表現を包含するために、これらの最小公倍数である1小節48分割の基本クロックを用いた。

FMC<sup>3</sup>のバージョン1で採用したのは6種類のドラム楽器、バスドラム・スネアドラム・クローズドハイハット・オープンハイハット・ライドシンバル・ハンドクラップである。個々の基本クロックのタイミングごとに、さらに3種類のスタイルごとに生成確率の重み付けを変えて(この匙加減がアルゴリズム作曲におけるノウハウ)、ONかOFFか、すなわちそのタイミングで打楽器を叩くのか何もしないのか、をいちいちサイコロを振って決定している。FLASH評価実験のヒアリング検討から、音楽的専門知識の無い一般のデザイナーでは、具体的な打楽器サウンドを時間軸にマップした一種の楽譜を作るのは困難なので、この部分はシステムお任せの自動生成とした。図4はバスドラムのパターン生成パッチの一例である。

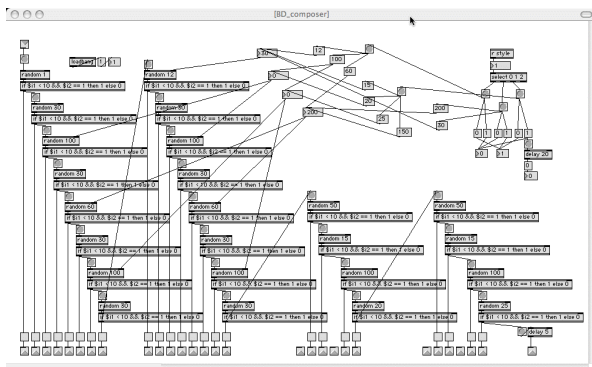


図4 バスドラムパターン生成パッチの一例

これら打楽器ごとのON/OFF情報は、パラメータ(m=76)からパラメータ(m=99)までの24地点(タイミングは1小節を0から47まで等分割した中から、順に「0 3 4 6 8 9 12 15 16 18 20 21 24 27 28 30 32 33 36 39 40 42 44 45」の地点だけを抽出して情報圧縮)のそれぞれにエンコードした[28]。

また、パラメータ(m=69)に「Drum Magic」という変数を定義し、当面は値ゼロとした。これは将来的にあと35種類の異なったバージョンとして、打楽器の種類やビートの種類を変えた、別のドラムセクションを実現するための拡張予約パラメータである。

#### 4.7 ベースパートの自動生成

FMC<sup>3</sup>では「大活躍するBass」というイメージを軸に、ベースパートでは「スケール」(個々の時間における音高方向の離散的な配置情報)によって、基本的なフレーズを自動生成する。FLASH評価実験

のヒアリング検討から、音楽的専門知識の無い一般のデザイナーでは、個々の音を配置した楽譜を修正するのは困難なので、内部的にはシーケンスデータを表示して修正できるサブ画面を持っているが、バージョン1ではこの画面は敢えて開かないようにした。コードを構成する重要なノートとリズム的に重要なタイミング(ビート)を基幹として、そこから低い発生頻度での逸脱を許可(テンションノートとして許容される)しつつ、時間的關係性による配置という要素を盛り込んだ生成アルゴリズムを実装した。

2小節単位のベースパターンを繰り返す時間軸方向では、16ビートとシャッフルの同時演奏を排除して最大16ステップ(シャッフルの場合には先頭12ステップを利用)とし、パラメータ(m=36)からパラメータ(m=67)までの32文字でベース2小節全ての演奏情報を乱数により選択・格納し、ドラムパートと同様に無駄のない情報圧縮を行った。

音高方向についてはスケールを検討し、36データを「3オクターブ+5度」の範囲に割り当て、内部的にこれらのパラメータをMax/MSPのtableオブジェクトに格納(休符=無音は音高ゼロと定義)した。それぞれのtableにおいて、与えられた乱数を引き数としてベースの音高をテーブルが返すので、データの幅(占有率)がそのノートの生成確率となる。さらにパラメータ(m=68)の1文字の36ステップを「ベース音の生成密度」として利用して、テーブル参照を行う確率として作用させた。これにより、ベースがたまたま同じ音列(メロディー)となっても、時間あたりのノート数の分布が大きく異なり、生成されるベースパートの単調さを解消する効果を得た。図5はベースのパターン生成パッチで参照するテーブルの一例である。

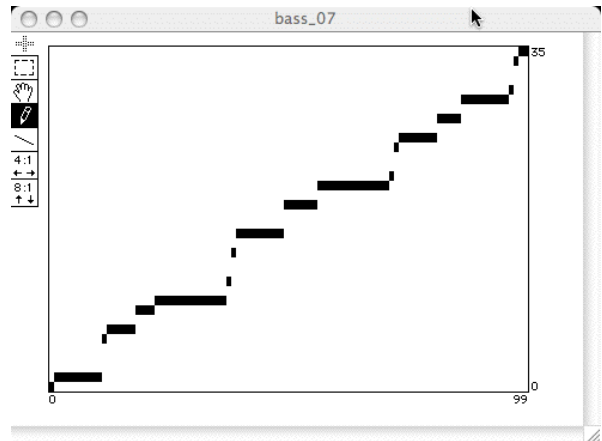


図5 ベースパターン生成テーブルの一例

#### 4.8 コード進行の生成

FMC<sup>3</sup>では、基幹となるドラムとベースに加えて控え目な音量で「アルペジオ」と「カッティングコード」を加えることにより、和声(コード)の感覚を音楽クリップに持ち込んだ。FMC<sup>3</sup>の生成する音楽クリップは、基本的にループを繰り返すミニ



マルミュージックの一種と言えるが、その単調さを解消する最大の要素は、コード進行が「バリエーションを持ち」「意外性も加味し」ながら、しかし同時に「予定調和的に(現代音楽のような難解さ/不自然さを回避して)」展開するところにある。ここでは局所的な暫定解決感、すなわち広義のドミナントモーションを指導原理とした。

機能と声とを12+12の調性に拡張したBachの音楽の即興パート、あるいはBachを敬愛するミュージシャンの多いJazzでのインプロビゼーション等において、近親調や平行調とは別に、「和声進行」「転調」で12平均率のどんな調/和音にいきなり飛んでいって(とりあえず驚いても構わない(ような音楽的解決法が存在する)。転調、あるいは和声進行したその瞬間は非常にテンションの高い(違和感、不協和感)状態であっても、「11m7 V7(1)」あるいはドミナント7thコードからの半音下行(V7 1の裏進行)があれば、聴取する人間は局所的な解決(トライトーンの解消)を知覚認知し、この短期記憶は和声進行に関する長期記憶まで後付けのように再解釈させ、最終的な解決に至るテンションであったと良好な印象をもたらす。FMC<sup>3</sup>のコード進行生成アルゴリズムにおいては、このBachとJazzに共通する経験的知見を活用した。

FMC<sup>3</sup>のコード進行生成ブロックのバージョン1では、パラメータ(m=8)とパラメータ(m=11)から生成される計72パターンのうち53種類(残りは将来的な拡張用にreserve)を設定して、以下のような4小節パターンを用意し、そこから自動作曲のたびにランダム選択することとした。

- (1) Jazzコード理論の教科書にあるようなドミナントモーションの組み合わせ
- (2) 古典和声教科書にある近親調の範囲での和声進行
- (3) 「循環コード」のようによく用いられているコード進行
- (4) PopsやRockで定番となっているコード進行

FLASH評価実験のヒアリング検討から、音楽的専門知識の無い一般のデザイナーの大部分は「コード」「和声進行」の知識を持たないので、本システムでは多数の妥当な候補からの選択とした。

FMC<sup>3</sup>のバージョン1の53パターンのうち24パターンを占めるドミナントモーション系のコード進行パターンには、単純なルールの組み合わせアルゴリズムで全ての場合を実装した(詳細は[28]を参照)。コードネームの値はその4小節の基調(tonal center)に対する相対的なrootの音程(12の剰余)とし、+12することで「minor7thコード」、12未満は「ドミナント7thコード」とした。常に進行を続けるために「Tonic」(終止コード)が不要なのも本システムのポイントである。

4小節パターンで回るFMC<sup>3</sup>のコード進行は、生成された音楽クリップごとに一定である。その単調さを打破するのは、4小節ごとに設定されている基調を「次の4小節ではrootを12音のいずれかからランダムに選んで移動する」という強制転調パラメータで、パラメータ(m=3)からパラメータ(m=6)までの4文字(4小節単位で4回の転調設定)の値(0-

11)として定義した。値は直前の4小節パターンの最後のコードのrootに対する相対的な音程(12の剰余)で、これが次の4小節の基調となる。転調は4小節ブロックごとの相対的な音程移動なので、16小節後に元に戻る確率は非常に少ない。12音のどの音程関係に転調しても、その後のコード進行によって瞬間的な違和感がやがて解決に至るテンションという印象を持つように知覚され、「生成パラメータは単純なのに次々と展開する音楽」「コード進行を書き留めてみると既存の楽曲にはおよそあり得ない(ことも多い)」という効果を得ることができた。図6は、「Advanced Control Panel」に表示された、4小節単位のコード進行パターン(その左側が転調パターン)である。

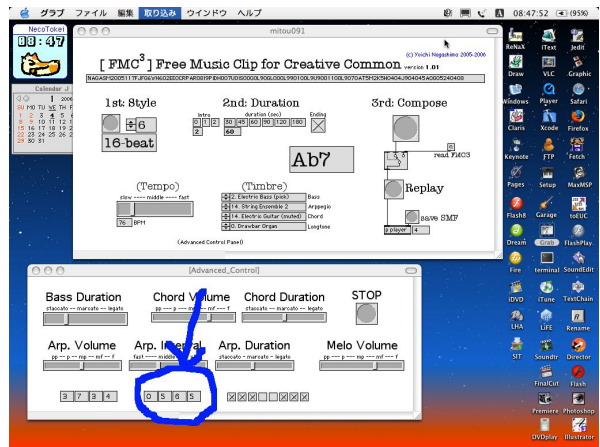


図6 コードパターン表示の一例

#### 4.9 アルペジオパートの生成

上記アルゴリズムによってコード進行が生成され、これを用いてFMC<sup>3</sup>では、メインのドラムとベースに加えて、控え目な音量で和音による伴奏を付加する。ただし基本方針として「和音の構成音を全て同時に鳴らす」というダサい演奏は避けたいので、コードを構成する個々の音を時間的にずらして鳴らす「アルペジオ」とした。

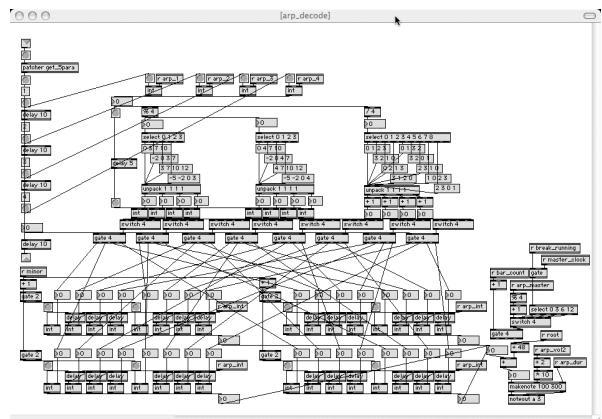


図7 アルペジオ生成パッチの一例



図7はその生成パッチで、分散和音として5文字分の生成パラメータによって伴奏情報を生成する。パラメータ(m=31)として定義したのは「各小節の1拍目のどこからアルペジオを開始するか」というスタートポイントで、1拍目を16分音符単位に4分割し「1拍目の最初/16分音符遅れ/8分音符遅れ/2拍目の最初」という4箇所として、値を「パラメータ(m=31)を4で割った剰余」として加算する。ここに、ドラムとベースに対する9段階の相対的音量を「パラメータ(m=31)を4で割った商」として加算し、計36パターンとして規定した。

FMC<sup>3</sup>パラメータ(m=32)からパラメータ(m=35)として、ループを構成する4小節それぞれに対して、4和音である分散コードの転回系(4種)を「パラメータを4で割った剰余」、分散和音の形状(9種)を「パラメータを4で割った商」として、全ての可能性を盛り込んだ計36パターンに定義した[28]。

このようにFMC<sup>3</sup>のアルペジオパートは、小節ごとに異なるコードに対して、分散和音の展開系(ボイシング)、さらに分散和音の音型までそれぞれ別個に演奏していて、電子キーボード等にある単純なパターンの繰り返しに比べて複雑なパターンを演奏する。パターンの先頭ポイントが16分音符単位で遅れる仕様も、ヒューマンな味を出すための経験則から来たテクニックである。

また、パラメータ(m=27)の"Arp. Interval"として、アルペジオパートの音同士の時間的間隔を"Advanced Control Parameter"に定義した。これはパラメータ(m=7)の「スタイル」を選択するたびに、選ばれたスタイルとテンポに応じた範囲内の適当なアルペジオ音オンセット時間差(音楽のテンポとは無関係の絶対的時間。ヒューマンな印象を演出できる)がランダムに設定され、変更が可能である。パラメータ(m=28)の"Arp. Duration"としては、アルペジオパートDuration(スタッカートやレガートの表現)も定義した。

#### 4.10 コード(バックング)パートの生成

コード進行に対応してFMC<sup>3</sup>で生成する音楽情報として、メインのドラムとベースに加えて、控え目な音量で「和音の一部を短くカッティング演奏」というパートも付加した。機能としてはコード(ドミナント7thとマイナー7thの2種)を構成する4音のうち、重要な2音だけを同時に「各小節の2拍目(必ず演奏)」と「4拍目(無し/演奏)」に2音フレーズとして短く鳴らす仕様とした。

パラメータ(m=70)からパラメータ(m=73)には、ループを構成する4小節それぞれに対して3種類のコード(バックング)パート生成情報を、3種類のピートスタイルと2種類のコードタイプごとにアルゴリズム生成し多重化(総計で2\*3\*6=36通り)した(詳細は[28])。Jazzの世界では「4和音のコードの全ての構成音を鳴らすのはダサい」というのは常識であり、このように演奏する音の数を限定することで、よりバリエーションを増やすメリットを得ることができた。

パラメータ(m=25)の"Chord Volume"としては、

カッティング・コードパートの音量バランスを、固定されたベースとドラムに対して設定できるようにAdvanced Control Parameterに定義した。さらにパラメータ(m=26)"Chord Duration"として、カッティング・コードの音の長さDuration(スタッカートやレガートの表現)も定義した。これも、パラメータ(m=7)の「スタイル」を選択するたびに、選ばれたスタイルとテンポに応じた範囲内の適当なカッティングDurationがランダムに設定され(変更可能)、この表現により、カッティングコードパートの「ノリ」の印象を大きく変化させることができる。図8はカッティング・コードパートの生成ブロックのサブパッチである。

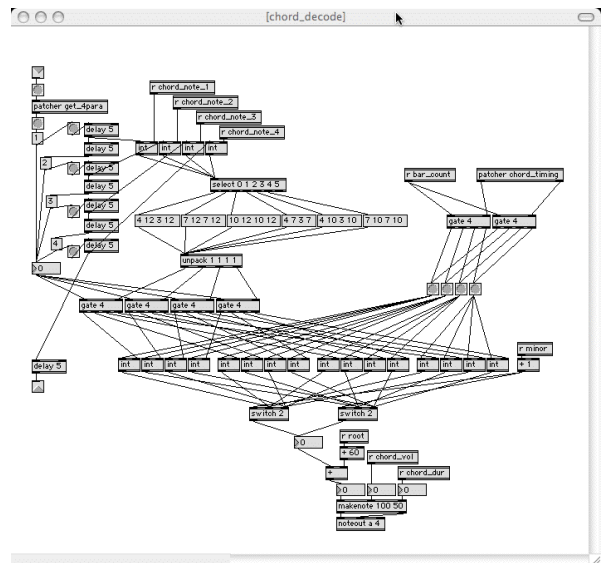


図8 コード(バックング)生成パッチの一例

#### 4.11 ロングノートパートの生成

FMC<sup>3</sup>の開発において最後に付加したのが、小節ごとに2音のロングトーンを「疑似メロディー」のように演奏するパートである。この生成アルゴリズムを一言で言えば、「コード(min7thコードまたはドミナント7thコード)を構成する4音に加えて、UST(上部構成3和音)として許容されるテンションノートを候補に追加する」という事である。

まずパラメータ(m=16)からパラメータ(m=19)を1-4小節目に対応させ、ロングトーン・フレーズのリズムパターンを、4小節それぞれに36パターン定義した。次に、パラメータ(m=20)からパラメータ(m=23)を1-4小節目に対応させ、ロングトーン・フレーズのノート(音高)情報を、その瞬間のコードのrootに対する音程として、バージョン1では4小節それぞれに30パターン(全ての組み合わせ)を定義した。このアルゴリズムの詳細はFMC<sup>3</sup>公開サイト[28]に譲るが、その中核は以下となる。

割り当てるノートの候補のrootに対する音程は、「4音からなるコードの構成音：7度/8度/10度/12度」+「その上部構成3和音(UST)：9度/11度/13度」である。ここからavoid noteを除外して、

「ドミナント7th + 9度 + 13度」「マイナー7th + 9度 + 11度」の組み合わせを全て網羅する。2種類のコードネーム(ドミナント7th、マイナー7th)ごとに、第3音(10度)に相当するノートはrootに対して半音offsetで「15/16」を切り替える。2種類のコードネームに、第3音以外に割り当てるノート候補は共通に、7度(10)、8度(12)、12度(19)、9度(14)、11度(17)、13度(21)とする。

以上のルールだけでは「必ずロングノートが降り続ける」という不自然な音楽になるので、パラメータ(m=30)として"LongNote Density"を定義した。これは、ロングノート・疑似メロディーパートを、8小節を単位として「それぞれの小節で演奏するかしないか」という36種類の16進パターン「88 44 22 11 cc 66 33 aa 55 B6 da 9a 9c 9e 9f ee dd bb 77 cd cb c7 ce cf b7 bf bd be e7 eb ec 99 96 94 8c 5a」として設定する。例えばパターン"B6"であれば2進数の"10110110"、つまり8小節ループのうち1,3,4,6,7小節でだけ、上記アルゴリズムに対応したリズムとノートに対応した演奏を付加する。パラメータ(m=29)は"LongNote Volume"として、ロングノート・疑似メロディーパートの音量バランスを設定する。

## 5. FMC<sup>3</sup>の評価と今後の課題

### 5.1 関連ユーティリティ

FMC<sup>3</sup>関連ユーティリティとして、(1)MIDIファイル16進ダンプユーティリティ"hexdump"、(2)標準MIDIファイルFMC<sup>3</sup>パラメータ埋込みユーティリティ"NAGASM"、(3)FMC<sup>3</sup>パラメータ抽出ユーティリティ"paramcheck"をMacOSX版として公開した(Windows/Unixについては公開したCソースプログラムから作成可能)[28]。

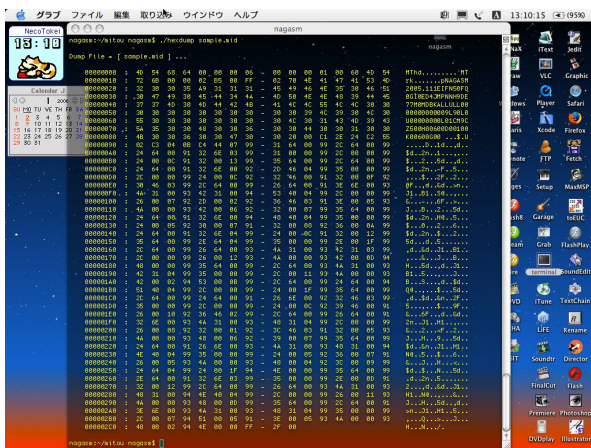


図9 ユーティリティで埋め込んだパラメータ例

(2)のツールは、標準MIDIファイルの冒頭にplain textのFMC<sup>3</sup>パラメータを埋め込むもので、「./NAGASM target\_SMF param\_TXT」のコマンドで、"target\_SMF"(拡張子.mid)の中に、param\_TXT(FMC<sup>3</sup>パラメータファイル、拡張子.txt)

の内容を埋め込んで、同じターゲットファイルに上書きする。図9はこのツールによりFMC<sup>3</sup>パラメータを埋め込んだ標準MIDIファイルを、HEXダンプユーティリティによって表示した例である。パラメータはMIDI演奏情報の記述とは関係ない付加情報部分と定義して埋め込まれているので、普通のMIDIシーケンスソフトでも問題なく取り扱い出来る。

### 5.2 FMC<sup>3</sup>の評価と応用例

本研究では、システムの開発と平行して実際にサンプル音楽クリップを試作生成しFMC<sup>3</sup>公開サイト[28]に置いた。これとともに、具体的にFLASHな作品系コンテンツに利用してもらってコメントやアドバイスを求める、という広義の評価実験も行っている。この結果は引き続きシステムの改良に反映させつつ、広く公開して議論検討していく予定である。

実際にFMC<sup>3</sup>を使ってコンテンツのサウンドトラックを生成したユーザからの感想として、「いくらでも続く」「次のパターンが予測しにくく飽きがこない」「それなりにいい感じでビートに乗れる」というような生成音楽クリップへの報告も得ている。一例として、筆者の面識のないある大学院生が、国際会議シググラフの応募論文デモムービーのサウンドトラックとしてFMC<sup>3</sup>を利用した事例、FLASHゲームを制作する中でBGMとして簡単に利用できたという報告事例、などがある。

一方、過去の学会発表の質疑やコンテンツクリエイターなどとの議論の中で、単独の音楽としてもっともFMC<sup>3</sup>に欠けているものとして「メロディが感じられない(聞こえない)」という意見が圧倒的に多かった。これはある意味で当然のことで、設計当初よりメロディという概念を敢えて中心から避けてきたからである。

また、初心者ユーザをターゲットとするのであれば、もっと直感的に分かりやすいインターフェースが必要なのは、という評価もあった。DTMソフトウェアやGarageBandのような、トラックが並んだ画面は筆者の印象では中級者向けであると判断し、バージョン1のようにシンプルな画面にしたつもりであったが、これは今後の課題である。

### 5.3 今後に向けての考察

このFMC<sup>3</sup>の自動作曲アルゴリズムを組み込みシステムへ活用する発展系としては、自動車や店舗などでのBGM生成に活用する「いくら聞いても使っても無料のBGM」という可能性があり、今後に向けての検討課題の一つとしていきたい。

また、生成する音楽を主役でないBGMと割り切らず、「単独の音楽として聴取/認知できる」という自動生成アルゴリズムの新たな戦略の可能性も課題と考えている。ここでは、本システムで主役とせず避けてきた「メロディ」を新たな意味付けのもとで生成アルゴリズムの柱として統合してい

く、という新しいアプローチが必要となる。

さらに、バージョン1ではユーザから伏せているパラメータ設定/編集画面など、ユーザインターフェースの改良も重要なテーマであり、関連学会での議論などを交えて検討改良を進めていきたい。

## 6. むすび

「誰でも手軽に作品系FLASHコンテンツの音楽パートを自動生成するシステム」を開発した。本システムは最初のアプローチであり、多くのクリエイタ/デザイナーとの意見交換により、さらに内容の充実したシステムに成長させていく計画である。これにより、コンテンツビジネス、コンテンツ文化の隆盛に貢献できるように努力していきたい。

## 謝辞

本研究は、独立行政法人 情報処理推進機構 (IPA)の「2005年上期 未踏ソフトウェア創造事業 (竹林洋一PM)」に採択・助成[30]され行ったものであり、IPAおよび竹林PMに感謝いたします。また、本研究に関して有意義な意見・討論をいただいた、情報処理学会音楽情報科学研究会、日本音楽知覚認知学会、情報処理学会エンタテインメントコンピューティング研究会、浜松音楽情報科学研究会(HMACS)の皆様に感謝いたします。また、本研究にあたって、被験者・サンプル制作等で協力いただいた、静岡文化芸術大学の学生・院生の皆様に感謝いたします。

## 参考文献

- [1]柴田南雄・徳丸吉彦・他：コンピューターと音楽、カワイ楽譜、1972
- [2]西村恕彦・音楽情報科学研究会：コンピューターと音楽、共立出版、1987
- [3]情報処理学会誌：特集「計算機と音楽」、情報処理、Vol.29 No.6、情報処理学会、1988
- [4]Leonard B. Meyer：Style and Music, The University of Chicago Press, 1989
- [5]F. Richard Moore：Elements of Computer Music, Prentice Hall, 1990
- [6]Leonard B. Meyer：Emotion and Meaning in Music, The University of Chicago Press, 1992
- [7]平成4年度文部省科学研究費総合研究(B)：音楽情報処理の技術的基盤「音楽情報科学に関する総合的研究」、課題番号04352030、1993
- [8]Leonard B. Meyer：Music, the Arts, and Ideas, The University of Chicago Press, 1994
- [9]情報処理学会誌：特集「音楽情報処理」、情報処理、Vol.35 No.9、情報処理学会、1994
- [10]Curtis Roads：The Computer Music Tutorial, The MIT Press, 1996
- [11]長嶋・橋本・平賀・平田：コンピューターと音楽の世界、共立出版、1998
- [12]Charles Madden：Fractals in Music, High Art Press, 1999
- [13]長嶋洋一：コンピューターサウンドの世界、CQ出版、1999
- [14]David Cope：Virtual Music - Computer Synthesis of Musical Style -, The MIT Press, 2001
- [15]Curtis Roads - 青柳・小坂・平田・堀内・他：コ

ンピュータ音楽 - 歴史・テクノロジー・アート - , 東京電機大学出版局, 2001

- [16]ICMA <http://www.computermusic.org>
- [17]ASL <http://nagasm.org>
- [18]Y.Nagashima：“IMPROVISESSION-II”：A Performing / Composing System for Improvisational Sessions with Networks, Proceedings of International Workshop on Entertainment Computing, 2002
- [19]Y.Nagashima：“GDS (Global Delayed Session) Music - new improvisational music with network latency, Proceedings of 2003 International Computer Music Conference, ICMA, 2003
- [20]長嶋洋一：音楽的ビートが映像的ビートの知覚に及ぼす引き込み効果、芸術科学会論文誌 Vol.3 No.1、芸術科学会、2003
- [21]Y.Nagashima：“Drawing-in effect on perception / cognition of musical beats and visual beats, Proceedings of International Symposium on Musical Acoustics, ISMA, 2004
- [22]長嶋洋一：作品系FLASHコンテンツの分類と傾向について、情報処理学会研究報告 Vol.2005, No.59 (2005-EC-1)、情報処理学会、2005
- [23]長嶋洋一：作品系コンテンツのための自動作曲システムに向けて(1)、情報処理学会研究報告 Vol.2005, No.82 (2005-MUS-61)、情報処理学会、2005
- [24]長嶋洋一：作品系コンテンツのための自動作曲システムに向けて(2)、情報処理学会研究報告 Vol.2005, No.100 (2005-MUS-62)、情報処理学会、2005
- [25]長嶋洋一：作品系コンテンツのための自動作曲システムに向けて(3)、日本音楽知覚認知学会秋季研究発表会資料集、日本音楽知覚認知学会、2005
- [26]長嶋洋一：作品系コンテンツのための自動作曲システムに向けて(4)、情報処理学会研究報告 Vol.2005, No.129 (2005-MUS-63)、情報処理学会、2005
- [27]著作権法 <http://law.e-gov.go.jp/htmlldata/S45/S45H0048.html>
- [28]FMC<sup>3</sup> <http://suac.net/FMC3/>
- [29]クリエイティブ・コモンズ・ジャパン、<http://www.creativecommons.jp/>
- [30]<http://www.ipa.go.jp/jinzai/esp/2005mito1/koubokekka.html>