

長嶋 洋一
デザイン学部技術造形学科

Yoichi NAGASHIMA
Department of Art and Science, Faculty of Design

メディア心理学研究、およびエンタテインメントコンピューティング研究の対象として、Web コンテンツの一つであるFLASHと音楽をテーマとした。目標を「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」の開発・公開とし、名称を「FMC³」(Free Music Clip for Creative Common: クリエイターのためのフリー音楽素材)と決めた。既存の多数のFLASH作品の検討/分類/評価実験を受けて、ショートムービーやフラッシュなどの作品系コンテンツの特性を音楽的に分析してシステムの仕様を検討し、著作権の問題をクリアして将来的にフリー公開する手法もあわせて検討した。システム開発には段階的なステップを計画して、中核となるアルゴリズム作曲エンジン部分を心理学実験により改訂・成長させる余地を確保した。本稿はこの研究報告の前編である。

This report is a part of the research of the system that composes the music part to produce the artistic contents automatically. I collected a lot of FLASH works, classified those features, and discussed the tendency. The next step is to develop the "algorithmic composition" engine and I discussed with musicology.

1. はじめに

筆者はこれまで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¹⁾。その中で、音楽でもインスタレーション(体験参加型造形作品)でもパフォーマンスでも、システムと人間との関係におけるエンタテインメント性の視点が重要であることに注目し、関連した研究テーマとして、ネットワークを利用した音楽セッションシステム²⁾³⁾、マルチメディアコンテンツにおける映像と音楽のビートの同期について⁴⁾⁵⁾、などに取り組んできた。

そして今回、メディアデザイン教育を行っている多くの学生の熱烈な注目/支持に触発され、エンタテインメントコンピューティング研究のテーマとして、Web コンテンツの一つである「FLASH」と音楽に関連する領域を取り上げた。本研究の具体的な目標を「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」の開発・公開とし、このシステム開発は2005年前期IPA「未踏ソフトウェア創造事業」にも採択された⁶⁾。

研究は、予備的調査として既存の多数のFLASH作品を収集し、音楽パートに注目して作品の特徴を検討して分類し、「作品系FLASH」の傾向について検討することからスタートした。実際のシステム開発には段階的なステップを計画して、中核となるアルゴリズム作曲エンジン部分を心理学実験により改訂・成長させる余地を確保した。本稿が出版される時期には最終成果の公開が完了している予定であるが、誌面と原稿執筆時期の関係で、本稿では「前編」として、この開発研究の前半部分を報告する。

なお、誌面の関係で「参考文献」および「図」をすべて省略したので、文末のwebアドレスを参照されたい。

2. 研究の背景と目的

2-1. 研究の背景-メディアコンテンツ制作環境とFLASH

マルチメディア系コンテンツの開発環境という視点から、筆者は過去の単行本の中でいくつかの具体例を検討紹介してきた⁷⁾⁸⁾⁹⁾。本項ではその中で重要な3件について整理する。

図1はSun社が提唱し世界中に普及したJavaの開発中の風景である。Javaは仮想マシンの採用によりplatform非依存の初めての成功例として、インターネット時代のキラコンテンツ第1号(全てのブラウザが標準プラグインに対応)となった。セキュリティと提供サービスとのトレードオフ、エミュレーションによる遅さなどの課題を抱えながらも、垂流でない「100% pure Java」のアプレットは、Javaサーブレットとともに今後も発展すると思われる。

図2はMacromedia社がJava以前から提供しているDirector(Webコンテンツとしては「Shockwave」)の開発中の風景である。Directorはもともと、複雑・大規模・高度なコンテンツを制作するためのオーサリングツールであり、C言語などのプログラミングスキルの無いデザイナーやアーティストの開発を支援するが、製品はプロ用ソフトとしてかなり高価である(Shockwave Playerプラグインはフリー)。広告業界などで有名な映像制作ソフトの「Motion Dive」自体もDirectorで開発されるなど、現在でもアプリケーション開発とコンテンツ制作の主流ツールである。

そして図3が、本稿の主役のFlashの開発中の風景である。歴史的経緯としては、元々インターネット向けに「軽いアニメーション」コンテンツを制作する「Future Splash」というソフトがあった。Directorはグラフィクスがbitmap起源で、パソコン用CDROMコンテンツはともかく、Web用データ(dcrファイル)サイズが大きい欠点があった。Future Splashはvectorグラフィクスで

サイズが小さいが、専用プラグインが敬遠された。そこで Macromedia 社は Future Splash を会社ごと買収し Future Splash Animator (Flash1)として発表、以後、Flashとして両者のプラグインを統合した Shockwave (フリー)を提供した。現在ではほぼ全てのブラウザが Shockwave (swf ファイル) 標準対応となり、Flashは一気にWebコンテンツ制作ツールの主役に躍り出てきた。

2005年4月には、静止画/ドキュメントベースの世界標準WebコンテンツであるAcrobat(PDF)を提供するAdobe社が、4000億円ほどでMacromedia社を買収する、というニュースが世界を驚かせた。Macromedia社はDirectorとFlashの機能を相互に高めて(互いの長所を重複して共有)いたが、この提携により、ますますタンイラクティブな、あるいはダイナミックなWebコンテンツの制作ツールとして進展することを期待したい。

FLASHコンテンツの多彩な表現と軽さ、そして「標準プラグイン」化したことで、パソコンだけでなく、携帯電話などインターネット対応の電子機器組み込み向け「Flashプレイヤーのモドルウェア」も登場した。Javaのエミュレーションによる遅さに対し、専用プレーヤにより「小さく・速い」FLASHは、これからJavaと並ぶ主流として、爆発的に普及する可能性を感じさせる。

2-2. FLASHコンテンツとサウンド/音楽

上記のJava、Director(Shockwave)、Flashに筆者が注目する第一の理由は、メインテーマである「音楽」との融合というメリットである。もう一つのメリットとして、Microsoftの秘密主義(度重なるセキュリティのトラブルの元凶)の対極である、仕様のオープン性にも注目したい。筆者はまだ日本語Java資料の無い時代に、sun社がWebで公開するドキュメントだけを頼りに、Javaバイトコードを逆コンパイルするツールを容易に開発できた¹⁰⁾。FlashについてもMacromedia社はこのオープン性を重視し、詳細なドキュメントをフリーで公開している^{11) 12)}。この190ページのPDF資料により、他社ソフトはデータをFLASHフォーマットで書き出すことができ、研究する立場としては、FLASHコンテンツを生データのまま詳細に解析できる、という利点が重要である。

音楽情報科学の領域では、音楽演奏情報を分析研究する際に、演奏された音楽音響を対象として解析しようとする、いきなり「音源分離」「ビートトラッキング」「フレーズジグ抽出」などの難問に直面する¹³⁾。そこで、MIDI音源のMIDI情報受信から発音までの遅れのばらつきに十分に配慮した上で、MIDI化された音楽演奏データを解析することが一般的である^{14) 15) 16)}。

マルチメディアコンテンツの視覚的情報と音響情報との関係を研究する上では、映像トラックとサウンドトラックとが別々のブロックに圧縮されているようなムービー/ストリームデータでは、DVテープを記録した際に起きる「音ずれ」¹⁷⁾のような同期性の問題も考慮する必要がある。

Flashはタイムラインをベースとしたデータ体系であり、MIDIファイルのように仕様が詳細に公開されている。そこで生データをそのまま分析することで、実験環境の時間的遅れや同期のずれを心配することなく、マルチメディア間の関係を正確に解析できる。筆者の研究対象の一つは

メディア心理学実験にあるので、提示データについて明確に定義されている実験をデザインでき、被験者データの解析部分に安心して集中できるのは、研究上、非常に大きなメリットであると考えている。

2-3. 研究の目的-「使える音楽」の提供

メディアアートの世界では、広く一般大衆が享受するマルチメディアの姿を、およそ3年前-5年前には、常にアーティストやクリエイターが先導してきた。2005年、愛知万博の多数のパビリオンで提供された大画面シアターの映像体験は、3年前までは一部の先駆的なアーティスト/クリエイターが実験していたが、これは3年後には一般家庭で楽しめるようになっていようであろう。本研究のテーマはここ1-2年は基本的にクリエイター(とその卵)のためのシステムであるが、近い未来、「クリエイション(作品創造)の大衆化」という時代的要請として一般化すると思われる。

あらゆる機器やシステムやサービスがインターネット環境になった現代、大衆はネットワークから与えられる、ゲーム、ムービー、フラッシュなどのコンテンツを楽しめる。既にホームページ/ハイパーリンクは「見る」「検索する」だけでなく、お手軽ツールでホームページやブログとして「作る」「発信する」時代である。そして3-5年先に求められるのは、一般大衆自身がショートムービーやフラッシュ作品などの、マルチメディア作品系コンテンツを「自分で作る」事である。具体的な事例として以下の3点を挙げるが、ここには共通の問題点がある。その解決が、本テーマの目的である。

事例1

カメラで撮影したフィルムをDPE屋で現像プリントしてアルバムに並べたのは過去の話。現在ではデジカメ(携帯電話)で撮影した多量の写真画像をCDRに保存し、家庭でプリントしたり、テレビに出して見たり、携帯音楽プレーヤの画面で楽しめる。次に欲しいのは、デジカメ写真を自動で手軽にスライドショー(転換エフェクト付き)にして、ついでに音楽も加えたムービー「作品」に作り上げて楽しむことである。既に“LIFE with PhotoCinema”のようなソフトウェアで、これは容易に実現できる。

事例2

ビデオカメラで撮影した映像を、高価な映像編集機(アナログのダビングは3回が限界)を使って、映像作品/ホームビデオを作っていたのは過去の話。現在ではパソコン付属の映像編集ソフトに取り込んで、ついでに音楽も加えたムービー「作品」として、かなりお手軽にホームビデオDVDが作れるようになった。Windowsであれば“ムービーメーカー”、Macであれば“iMovie”と“iDVD”という添付ソフトでこれは容易に実現できる。

事例3

インタラクティブなコンテンツを作る各種の環境の中で、「フラッシュ」は“Flash MX”の世代となって、Javaアプリを凌ぐ勢いで、携帯電話コンテンツやブラウザ標準対応プラグインとして定着してきた。従来の“Director”などの高度なオーサリングツールでなくても、多数のグラフィクスソフトが対応して、ついでにMP3形式の音楽を簡単に加えて、FLASH形式で書き出せる。新しい時代の、スライドショー、ムービー、プレゼンテーション、ゲームなどの標準である。

この3つの局面に共通するのは、コンテンツの視覚的要素の「お手軽創作」の支援である。その一方で、マルチメディアコンテンツとして両輪をなす筈の聴覚的要素、すなわち「サウンド」については、あまり進展していない状況にある。効果音などのサウンドは、ビデオカメラ（携帯電話）やMDウォークマンで録音でき、ショップやインターネットには多数の「著作権フリー・効果音集」がある。サウンドを切り貼りしたり残響など効果を加えるフリーソフトも充実している。しかし問題はサウンドのうち「音楽」である。市販CD等の音楽は、著作権により使えず、これが現在、最大のボトルネックなのである。

3. 「使えない」問題点の検討

本研究で対象とする「作品系FLASH」の制作において3年後の一般大衆、現在のクリエイター（とその卵）が熱烈に希望するのは、「使える」音楽データをお手軽に欲しい、という事である。もちろん、DTM（打込み音楽）ホビーストや音楽の専門知識のある人であれば、MIDI 機器やDTMソフトウェアによって、オリジナルの音楽を作って楽しむ環境は、ITの進展とともにとても充実している。しかし、いわば音楽の専門家でない一般大衆・ビジュアルデザイナー・クリエイターにとっても、マルチメディアコンテンツを創造するために、「使える音楽」が求められていて、これは現状では解決されていない。以下、現状でどう「使えない」のか、のうち主な3点について検討する。

3-1. 入手する—著作権のため使えない

いかに自分が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Webで公開する等の用途には無断では使えない。その楽曲を自分で「耳コピー」したり市販の楽譜を打込んでMIDIデータを作る、すなわちオリジナルアレンジであっても駄目である。現在、クリエイターの卵（コンテンツデザインの学生など）が行っている対策は、

- ・知り合いに作曲・提供してもらう
- ・著作権フリー音楽集を入手してその中から選ぶ
- ・音楽シェアウェア（有料で使用許諾された楽曲）を購入する

などである。その結果、「あの作品と音楽は同じじゃん」等のフラッシュ作品があちこちに生まれ、せっかくの創作が淋しいものになる。

3-2. 市販ソフト—いくつかの問題点

これまでに発表・市販されてきた多くのMIDIベースの「自動作曲」「作曲支援」ソフトの中にも、音楽的知識により個々の音符を打ち込むことなく「お手軽作曲」する、という音楽生成ソフトウェア製品は存在する。しかし「ポップス風」「カントリー風」等の別売ライブラリを買い揃えないと実用にならない。また、ミニマル自動生成系のソフトでは開発元のマニアックな思い入れからか機能/操作が複雑で、面白い楽曲出力を得るためには、グラフィックソフト以上の投資額と専門的知識が要求される。お手軽に音楽を欲しい立場にとっては、これまた、かなり敷居が高い状況である。

多数のサウンドフレーズ断片を選択して並べて繋げる、

というMac付属の“GarageBand”はシンプルで画期的な製品である。しかし実際にフラッシュ作品のサウンドトラックを制作してみると、多くのユーザが好むフレーズは膨大なライブラリの一部に集中して、バリエーション不足（同じような音楽の重複。発売されているオプションフレーズ集の購入が必要とより高くつく）の点で不満が残る。サウンドファイルのため、テンポ変更とか変拍子のような自由度に限界があり、時間的な編集性という点では大きな制約がある。

3-3. 研究者の自動作曲システム—面白くない

複数（有限個）の素材を用意し選択し接続していく、という手法には上記のような問題があり、自動作曲システムの本質的な指導原理としては「新たに音楽要素を生成する」というアルゴリズム作曲¹³⁾の手法（既におよそ50年ほどの歴史）が本命である。音楽情報科学の領域では、このようなアルゴリズム作曲による自動作曲の研究は、現在でも多くの大学等で細々と繰返されているが、その成果は惨澹たるもの（ポップでもなく現代音楽風でもなく芸術的でもなくただひたすら稚拙な音楽）で、使える成果は得られていない。その最大の理由は、「音楽的基礎/素養/情熱/勉強」の不足した工学者が、安易にIT応用のため皮相的に音楽テーマを選択している事にあり、音楽を分かっている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（少なくとも筆者の個人的な期待は20年近く裏切られ続けている）。

4. 作品系FLASHと収集分類

以上のような考察を受けて、2005年2月-7月に、インターネット上の既存の多数のFLASH作品を収集し、音楽パートに注目して作品の特徴を検討して分類し、有志学生を被験者として、ごく簡単な心理評価実験を交えて「作品系FLASH」の傾向についての検討を試みた¹⁸⁾。

4-1. FLASHコンテンツのタイプ

フリー百科事典ウィキペディア（Wikipedia）からFLASHコンテンツの種類（形態の例）に関する記述を見ても、

- ・アニメーション
- ・ゲーム
- ・ウェブサイトのナビゲーション
- ・音楽再生

という例が挙げられている（後述）。また、「FLASHの特徴に関する記述」（長所と短所）を見てみると、

- ・インタラクティブなサイトに向いている
- ・再生環境への依存度が低い
- ・画面サイズを変えても画質が劣化しない
- ・Webブラウザ等からテキスト検索できない
- ・Macromedia社以外の企業、個人でもFlashデータを加工、生成するソフトウェアを自由に開発、配布することができる。さらに特筆事項として、以下の2点の指摘も、本研究の分類においては注目すべきであろう。

FLASHの特徴1

インターネットコミュニティの参加者によって笑いネタ、時事ネタなどでジョーク的な内容のものが作られたりする。ただ、このようなFlashは著作権や肖像権（傾向として政治家が多い）を侵害していることも多々ある。日本において単に「フラッシュ」と呼ぶとこちらを指す場合もある

FLASHの特徴2

Flash作品の中には、一例として、最初に表示される画面は間違い探しなどの鑑賞者を画面に注視させる構成とし、次に突然画面が切り替わり、怪物が雄たけびの音と共に画面に表示される等の、鑑賞者を強く驚かせる事を意図された構成の作品もあり、特に電子掲示板にFlash作品への直リンクが書き込まれている場合は注意が必要である

4-2. 「作品系FLASH」と「除外したFLASH」

本研究においては「作品系FLASH」と対象を限定している。そこでまず第一に、「Webサイトのナビゲーション機能としてのFLASH」を対象から除外した。これはFLASHの目的がまったく違うからである。なお、ナビゲーションFLASHの例は、資生堂・ソニー・日産自動車など、日本の多くの企業サイトに見ることができる。

基本的にはこれ以外のWebページ（Flashプラグイン）で見ることができるものは、一応FLASHの「作品」（作家が制作した成果物という意味）である。ただし、それに加えて、本研究では以下のFLASH群も対象から除外したので、その理由とともにここに明記しておく。

▲どっきり系FLASH

上記「FLASHの特徴2」にあったタイプである。静かな絵や間違い探し・視覚の錯覚などで注視させ、突然に叫び声とグロ画像が出現するもので、その演出効果のための手法は興味深いが、被験者に強いストレスを与えるために対象から除外した。なお、「ホラー系」ストーリー作品は残した。

▲エロサイト系FLASH

除外しなかった作品系FLASHの中にも性的表現の入ったものは散見するが、ここで除外したのは、所謂エロサイト（18禁）の中にあるコンテンツである。インターネットビジネスは常にエロが牽引したという事実は社会的に興味もあるが、本研究においては無関係であり、女子学生も多い被験者にストレスを与えるために除外した。

▲グロサイト系FLASH

除外しなかった作品系FLASHの中にも暴力的/残虐的表現の入ったものは散見するが、ここで除外したのは、所謂グロ趣味サイトのコンテンツである。予想外に海外グロFLASHが多かったこと、中国のグロFLASHは特に容赦ない事など、文化人類学的に興味ある傾向もあったが、本研究においては被験者にストレスを与えるために除外した。

▲ゲームFLASH

ゲームというのは間違いなく一種の作品コンテンツであるが、本研究では、作品の仕掛けとしてインタラクティブ性を持つものは「作品FLASH」として残しつつ、所謂ゲームサイトのFLASHゲーム（得点や面クリアを目標とするゲームらしいゲーム）は対象から除外した。ゲームにも効果音やBGMはあるが、ゲームの攻略そのものが中心の目的である、という点で本研究の対象としては適当でないためである。

▲外国語能力の必要なFLASH

ストーリーと台詞や説明文章（字幕と会話音声の両方）のある作品系FLASHの中で、日本でなく海外のFLASHではその言語に慣れていないとまったく判らない、というものも多数あった。Motion Typographyなどイメージが伝われば判るものは別にして、語学に強くない被験者が多いため、これらの語学力を必要とする海外FLASHも除外した。

4-3. FLASHの収集作業

以上のような基準を整理した上で、2005年2月から7月にかけて「ネット上からFLASHをとにかく集める」作業を行った。Webには所謂「FLASH集サイト」が多数あるが、この中には上記で除外した「どっきり系」「エロFLASH」「グロFLASH」も多数、直リンクされており、収集作業を行った筆者は望まなくても全てからその直撃を食らった。

また、著作権や肖像権を侵害したり名誉毀損（誹謗中傷）ネタのために、作者名をペンネームにする、あるいはクレジットのまったく無いFLASH作品もとても多いが、これらはリンク先から消えて“Not Found”であるものが多かった。そこで、実験のためのFLASHライブラリとしてリンクを使用することを断念して、ソースのswfファイルを全てダウンロードして手元に保存した。

FLASH作者の中には、
・swfファイルを別途ダウンロード（お持ち帰り）用に置いて積極的に提供する
・フレーム等でswfファイル名を隠しブラウザで観るだけでソースを出したくない

という両極端の対応があった。後者の場合にも、HTMLソースを読んで追えば大抵はswfファイルを手元できるが、原理的にswfを取らせない幾つかの高度な手法もあり、このごく少数（全体の1%以下）の作品系FLASHのみ、リンクで対応した。

FLASHを作る作家/グループはアマチュアもプロも世界中に多数おり、またFLASH作品は制作まで1年かけた熟練の名作から「FLASHを始めて3日です」的な稚拙なものまで千差万別である。収集作業において閲覧したFLASHは7500本以上、その中で上記「対象外FLASH」「駄フラ」等をパスして、とりあえずswfファイルを手元に保存したFLASHは約4000本となった。FLASHのデータサイズは平均すると1本1MB以下であるが、実際には1本4KBから1本32MBまで個々のFLASHのサイズも千差万別であった。

4-4. 作品系FLASHコンテンツの分類

本研究で対象とした作品系FLASHの大部分は、Wikipediaの分類で言えば「アニメーション」と「音楽再生」ということになる。しかしここには、多くのタイプがあり、現在でも完全な分類に成功していない。作家の創造性と表現方法は無限であり、機械的に類型化することにはもともと限界がある。分類項目の一つとして「その他」は最後まで必要であると思われる。以下、作業の中で注目したいいくつかの代表的な「作品系FLASH例」を紹介するが、紙面の関係で詳細は省略する。

音楽系 FLASH

- MTV/ プロモ系 (cool)
- Motion Graphics (VJ) 系 (cool)
- Motion Typography 系 (cool)
- ジュークボックス系 (多ch サウンド活用)
- J-POP 等のカラオケ PV (歌詞付)
- 空耳音楽系・外国語/方言系 (笑系)
- ラップ系・会話音声の音楽化 (笑系)

物語系 FLASH

- 恋愛もの・冒険もの・歴史もの
- 生命もの (生・死・運命) (感動系)
- テーマもの (ホラー・和風・ご当地)
- ドキュメンタリー・Project X もの
- 不条理もの・不思議系・抽象芸術 (図 4)
- キャラもの (ゲーム・アニメ・2ch)

お笑い系 FLASH

- Motion Typography もの (セリフ+字幕)
- パロディー・コラージュ (図 5)
- 事件・政治家・被告等をおちゃらける
- 4コマ漫画的な連載/テーマもの

メッセージ/風刺/告発系 FLASH

- 政治テーマ・国際テーマ
- 市民運動テーマ・環境テーマ
- イベントの CM/ プロモ
- パソコンもの (Windows 非難系) (図 6)

ネットマンガ FLASH

- 紙芝居系 (静止画をクリックで進める)
- TV アニメの Web 版 (出版社が運営)

インタラクティブ系 FLASH (純ゲームを除く)

- 浮遊感覚もの
- シミュレーションもの

その他

- なお、本研究の場合には、
- ・「紙芝居」的に静止画をクリックして送るタイプ
 - ・セリフが字幕だけで出る (読ませる) サイレントムービー
 - ・サウンドがインタラクティブ性に附随した効果音響だけである作品
 - ・空耳系やパロディー系で対象となる音楽がテーマとして限定される作品群
 - ・演出効果として「無音」を意図したもの

など、制作においてオリジナル音楽を必要としないFLASHについては、最終的には対象から外れるという性格がある。ただし、研究の途中ではこれらも検討の対象として注目していきたい。

4-5. 被験者による予備的評価実験

前項の作品系FLASHの分類例は、あくまで収集作業において多数のFLASHに接した筆者の印象によるものであり、客観性も中立性もない。そのため、今後あらためて選抜したFLASH作品を材料としたメディア心理学実験を行うための予備的な作業として、メディアデザインとFLASHの表現に興味を持つ学生有志の協力を得て、保管したFLASHの中からさらに一部 (趣味的FLASH、対象外) をカットし、およそ1500本について、「FLASH 観倒し企画」として評価実験を行った。ここでは、被験者の感性でプラス・マイナスいずれかの点でピンときたものだけを任意にピックアップして、一言コメントを添えてもら

う、という形態を取った。その目的は、個人でなく複数の視点から多数のFLASH作品を鑑賞評価し、その中に共通点があるか、あるいはユニークで有効な視点を新たに発掘する、という事である。この結果については今後の機会に報告していきたい。

5. FMC³ (Free Music Clip for Creative Common)

5-1. 作品系 FLASH の音楽の特徴

本研究は「あらゆる音楽をなんでも自動作曲する」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スソフトやマニアックな自動作曲ソフトでは、あれも出来るこれも出来る、と「どのような音楽スタイルでも作曲できる」万能性・網羅性を指向することで、ある目的を持つ専門的でないユーザの希望から乖離している。

ここで対象と考えている、ショートムービーやフラッシュ作品のためのサウンドトラックとしては、

- ・全体/部分の時間的長さ (「尺」を固定)
- ・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・音楽構造に独特の傾向と要請がある。また、より高度なマルチメディアコンテンツ作品の創作において、グラフィックスのパートと音楽のパートとの同期は非常に重要な要素である。ここでは、

- ・サウンドのある部分だけを時間的に伸縮
- ・没入感/躍動感/幻惑感/浮遊感などの演出
 - －映像と音楽のビートを微調整
 - －「ノリ」「タメ」「グルーブ」

などの操作を駆使して演出効果を盛り込むことを指向するが、これは「GarageBand」のようなサウンドファイル形式のシステムでは困難である。

5-2. システム開発研究の基本戦略

以上の分析検討を受け、本研究では、「使えない」部分の課題に対して、基本的に以下のような戦略を検討・構想・実験して、「使える音楽データ」としての解決を目指した¹⁹⁾。

戦略 1

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は一切利用しない。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 (図 7)、容易にインターネット検索できるシステムとする。この手法を活用すると、既存の楽曲の特徴情報を同様にデータベース化することにより、本システムで生成された楽曲が既存の楽曲と天文学的確率で偶然に一致する、という偶発事後的な著作権侵害を未然にチェックする可能性を提供できる。

戦略 2

音楽データのアルゴリズム作曲自動生成は、基本的に MIDI 情報ベースで行う。これは上記の「作品系コンテンツ創作における特徴」に対応するためである。その出力は、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データ化するシステムと統合する。アルゴリズムに乱数 (不確定) 要素を持ち込む度合いについては、自動作曲の確率統計アルゴリズム

ム²⁰⁾中でユーザが選択できるようにする。

戦略3

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「唯一の主演」ではない。これは、ダンス系のDJ/VJの音楽や演劇/ミュージカルの音楽と共通する特性であり、イメージを一言で例示すれば「Drum'n Bass」である。この音楽は、ビジュアルあるいはダンス的な主演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没入感その他の演出効果を上げるが、骨格部分において、古典的な音楽の要素から、メロディー・和声・(リズム)を捨象している^{21) 22)}。

戦略4

楽曲であるからその誕生とともに音楽著作権が生まれるが、コンピュータのアルゴリズムによりいくらでも自動生成される多数の楽曲、それもメロディーらしいメロディーすら無いミュージッククリップに著作権があるのか、という社会的な問題提起をあわせて検討する。基本的姿勢は「フリー音楽クリップ」(コピー自由、改編自由、配布自由。制作者クレジットを希望し、無断販売のみ禁止)とする(後述)。

5-3. 対象プラットフォーム

本システムが開発対象とするプラットフォームは、Macintosh (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、関連したツール類をXcodeなどのツール(C++)で開発して連携させる。最終的な公開段階では、Max/MSPのソース(パッチ)まで全て公開する計画であるが、ユーザの手元に自動生成システムを分散させるのか、Webと連携したサーバ側に置いたシステムで自動生成するのか、という方針はまだ未確定である。

計算型(ノンリアルタイム)のアルゴリズム作曲システムの場合、コンピュータの能力により音楽データの生成時間の短縮は可能であるが、本システムではリアルタイム生成ということで、例えば90秒の音楽クリップを生成するためには必ず90秒以上が必要となる。多数の背景音楽素材を提供したい本システムにとって本質的なこの欠点については、「バッチにより多数のサンプルを自動生成・蓄積するシステム」を構築・稼動することで対応する。

5-4. ネーミング「FMC³」(クリエイターのためのフリー音楽素材)

音楽情報科学研究の領域に限定した話ではないが、新しいシステムの開発研究において重要なものの一つがネーミングである、と筆者は考えている。何度となく繰返し接することで、そのコンセプトが骨の髄まで染み込んで、開発研究そのものの推進力となる事を期待しつつ、本システムの名称を「FMC³」(Free Music Clip for Creative Common: クリエイターのためのフリー音楽素材、発音は「エフ・エム・シーキューブ」)と決めた。ここでは、以下の2点をポイントとして明確にしておきたい。

FMC³のポイント1

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、あくまで音楽素材であって、音楽作品ではない。音楽著作権を主張しない「Music Clip」であって、人間等が作曲した音楽作品「Musical Piece」ではない、というのが基本方針

である。従って、本システムを用いて制作した音楽クリップは誰でも自由に使用・改編できる。ただし、FMC³を元にさらにシーケンスソフト等を用いて改編(アレンジ)したものを「FMC³起源」というクレジット無しに自分の完全な著作物として宣言することは、著作権の考え方から見ると道義的に正しいことではない。出典のクレジットを添えればまったく問題ない、という立場である。

FMC³のポイント2

フリーである。本システムでは、図7のように生成パラメータをSMFデータ内に埋め込んでいるが、利用・改編・配布は全てフリーというのが基本姿勢である。自由にFLASHなどに積極的に活用して欲しいので、利用したい作家に余計な手間は求めない。なお、2004年頃から世界的に採用され始めた「Creative Commons」²⁰⁾とは、基本的に無関係である。筆者は「Creative Commons」の活動については大いに共感・協賛できるが、本研究については、成果物が楽曲でなく音楽素材である、という点から、正式にCCの枠組みに乗ることは敢えて避けている。FMC³の直訳は「創造的な共同体のためのフリー音楽クリップ」という事であり、名称の一部が似ているとすれば偶然である。

6. FMC³生成と動作の詳細

6-1. FMC³生成の基本フロー

FMC³生成のための本システムの基本フローは、図8のスクリーンショット中の上部のメイン画面(試作段階)によって容易に理解できる。図の下端にある2つのウィンドウは上級者・開発者のための「隠しウィンドウ」であり、最終的には一般ユーザは開くことができない仕様とする可能性も検討している。現状(試作中)のFMC³生成システムのメイン画面にあるのはたった3つのステップであり、

- ・第1ステップ-スタイルをランダムに選ぶボタンを押す
- ・第2ステップ-音楽クリップの「尺」(イントロ/エンディングの有無)を選択
- ・第3ステップ-「Compose」ボタンを押すと1-2秒でパラメータ生成(自動作曲)

というだけである。当然ながら、オペレーションに音楽的知識は要求されない。これで生成された音楽クリップはその場でリアルタイム演奏され、希望すれば「SAVE」ボタンにより、ファイル名を付けた標準MIDIファイル(*****.MID)として保存できる。

6-2. FMC³生成の第1/第2ステップ

まず最初に第1ステップでユーザが設定するボタンをクリックすると、スタイルが36種類の中からランダムに選ばれる。この36種類の内訳は、さらに3種類のブロックに分かれており、

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

というようにリズム/ビートの基本単位が選択される。この3種類ごとにそれぞれ12段階の「ベースラインの音候補」が配備されている。ベースラインの音候補とは、もっとも単純なものはコードのルート音(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、全音階 Whole Tone Scale、などで構

成されている。この「スタイル」、すなわち3種のリズム/ビートと12種のベース音候補、という36通りのパラメータは、後述する生成パラメータのうちの第7パラメータとして1文字にエンコードされる。

これに続く第2ステップでは、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選択する。対象が本格的な音楽作品でないために、基本的にはミニマルミュージックのように同じパターンを繰り返す、というコンテンツの特性に対応し、その一方で、全体の長さだけはアバウトに指定できるようになっている。具体的には、

- ・イントロ (長・短・無)
- ・エンディング (有無)
- ・30秒から180秒まで6段階の「尺」

として指定する。 $3 \times 2 \times 6 = 36$ 通りのこのパラメータは、後述する生成パラメータのうちの第9パラメータとして1文字にエンコードされる。

FLASH作家の大部分が音楽のイントロやエンディングはクレジットやローダと重複させる事から、「尺」はイントロとエンディングを含まない本体部分である。「尺」と「テンポ」の指定により、切り上げた2小節単位での演奏所要時間 (duration) が決定され、その前後にイントロとエンディングが付加される。FMC³の特性から、テンポは曲ごとに設定され、その演奏中のテンポは一定である。図9は、システムクロックを生成するブロックのパッチであり、ドラムやベースのタイミングクロック、イントロ、エンディング、アルペジオパート、コード(バックング)パートなどのタイミング信号、1小節単位でのコード進行のチェンジ、4小節単位での基調移動(転調)のタイミングなどを生成している。

基本的にはこの2ステップで自動作曲に進めるが、第1ステップのボタンでは同時に、4/4のメトロノーム速度で50から200程度の範囲に分布する36段階の値の一つとして、第10パラメータの1文字にエンコードされる。このテンポは画面内にかなりアバウトに「slow-middle-fast」と指定されたスライダー範囲によって表示されており、希望すればこのスライダーを操作することで、後からテンポ・パラメータを設定変更して、自動作曲ステップに進むこともできる。

さらに第1ステップのボタンでは同時に、各パートごとの楽器音色もランダムに選択して、第12-第15パラメータの1文字にエンコードされる。この楽器音色名も画面内に表示されており、希望すればアップダウンボタンを操作することで、音色を設定変更して自動作曲ステップに進んだり、音色やテンポを変更して「再度演奏」(パラメータ再設定)に進むこともできる。

市販の自動作曲ソフトウェアであれば、ここからコード、キー、伴奏パターン、リズムパターン、あるいは実際のフレーズの一つ一つの音情報などを「打ち込む」ことになる。しかし本システムではここでユーザの入力は終了して、あとは第3ステップの「作曲スタート」のボタンしかない。これをクリックすると、もうSMFデータ、対応したMP3サウンド、という形で新しい音楽クリップがこの世に誕生する。「尺」と「テンポ」はユーザがFLASHや映像作品を作るために指定したい「演奏時パラメータ」であるが、その他を全て「生成用パラメータ」として隠匿し

てしまう点が、「音楽知識を必要としない」本システムの重要なポイントである。

6-3. FMC³生成のためのパラメータ

FMC³の生成パラメータ数は、「1行メールとして送れるサイズ」を念頭に、FMC³の全体として「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量は

- ・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進法)ので、生成される音楽クリップごとに全てのパラメータが完全に完全に異なる場合には、理論上「36の100乗(6の200乗)」の組み合わせがある事になる。実際には一部をFMC³の管理情報(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90乗」程度とするが、それでも十分なバラエティを提供できると考えている。

図10は、この「100文字パラメータ」を扱うブロックのパッチである。ここでは

- ・自動作曲エンジン部で生成されたパラメータをテーブルに保存
- ・外部のテキストファイルからパラメータを読み込みテーブルに格納
- ・パラメータファイルをテキストファイルとして書き出し
- ・パラメータ列を画面のテキストエリアに表示

などの処理を行う。Max/MSPは内部的にリアルタイム・アクセスできるデータテーブルを任意に定義できるので、データ幅100ポイント、データ量「0-35」というテーブルとして、FMC³の生成パラメータを内部的に格納した。デバッグ/管理者モードにおいてはこのテーブルをグラフィックに表示できるので、自動作曲の様子が可視化されており、この部分の可能性は今後、「音楽生成情報の可視化」という、新しいテーマとして検討することとした。

本システムでは音楽データの自動生成に先立って、まずアルゴリズム作曲エンジンブロックでこの生成用パラメータ群を乱数を用いて自動生成し、演奏パラメータである「尺」「テンポ」などととともに、「nagasm.txt」(予約名)というplain textファイルに書き出す。このファイルは先頭に「NAGASM2005」という10文字の加わった110バイトのデータであり、plain textファイルとして1曲のパラメータが1行100文字、という統一形式で処理できる。将来的に何10万曲という単位でデータベース処理する際にも、この軽さは有効であろう。

6-4. FMC³生成の自動作曲エンジン部分

メイン画面における第3ステップでトリガされるのが、本システムの中核とも言える「アルゴリズム作曲エンジン」部分である。図11はそのパッチであり、個々の音楽的要素は全てサブパッチとして階層化されている。図に従ってその動作を追えば、

- ・バスドラムのパターン生成
- ・スネアドラムのパターン生成
- ・クローズドハイハットのパターン生成

- ・オープンハイハットのパターン生成
- ・ライドシンバルのパターン生成
- ・ハンドクラップのパターン生成
- ・ベースの1小節目のフレーズ生成
- ・ベースの2小節目のフレーズ生成
- ・4小節のコード進行生成
- ・4小節ごとの基調の移動パターン生成
- ・アルペジオ伴奏パターン生成
- ・コード(バックング)パターン生成
- ・ロングノート(メロディー)のフレーズ生成
- ・イントロの生成
- ・エンディングの生成

という処理を次々に行い、終了すると出力トリガを出す。なお、本稿執筆時点ではこの中の一部ブロックは未実装である。

6-5. FMC³のドラムパート生成アルゴリズム

まず仕様として「36ステップの変数100文字」と限定した下で、本システムでは基幹となるドラムとベース(一種のフレーズ)の生成に重点を置いた。FMC³ではすべての音楽クリップは4小節単位で進行し、キーが小節単位で移動(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すし、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。図11の自動作曲エンジン部分のパッチでは、ドラムとベース以外の音楽的要素は単にパラメータ(1文字あたり36通り)のランダム指定を行うだけであるが、ドラムとベースは以下のように詳細にベース音高情報/ドラム楽器情報とその発音タイミングを持つ演奏情報をパターンとして生成し、パラメータとして管理する。

ベースとドラムのリズムの分解能は、1小節を8分割した「8ビート」、16分割した「16ビート」、12分割した「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「48」分割する基本クロックを用いることにした。これは、昔からMIDIの世界で定番の「タイムベース=48」そのものである。

図12は、6種類のドラム楽器のうちバスドラムのパターンを生成するサブパッチの例であり、同様のパッチが「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」用にも設置されている。個々の基本クロックのタイミングごとに、さらに「8ビート/16ビート/シャッフル」というスタイルごとに生成確率の重み付けを変えて、ONかOFFか、すなわちそのタイミングでバスドラムを蹴る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。開発中のアドバンスモードでは、この決定された個々のタイミングのON/OFFを手作業により変更可能ともしているが、これは一般的なシーケンサによる「打ち込み」そのものになってしまうため、この仕様を公開するかどうかはまだ未定である。

分割した音符に乗っていないタイミングでの演奏は現実的には「とても聞けない」ものであり、1小節を48分割したタイミングの全てに打楽器情報を割り当てるのは、休符の情報ばかりが多くて無駄である。そこで打楽器の鳴ら

ないビートを抜いた24種類のタイミングのみで演奏パッチ側の対応マトリクスを形成した。これにより、ドラムパターンを第76パラメータから第99パラメータまでの24文字で全てを格納することができる。

また、第69パラメータの1文字に「Drum Magic」という変数を定義し、当面は値ゼロとして設計した。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。

6-6. FMC³のベースパート生成アルゴリズム

本システムは「Drum'n 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、まず基本的なフレーズを生成する。これは例えば、バックングでCm7とかDm7/Gというコードが鳴っている時に、ベースなりピアノなりが、あるピッチクラスからランダムに選ばれたノート(一例:C,D,Eb,F,G,A,Bb,B)を、基本単位(16ビートなら1小節を16分割、シャッフルなら12分割したいずれかのタイミング)のいずれかで鳴らしていれば、なんとなくアドリブソロ(フレーズ/リフ)を演奏しているように聞こえる、という作曲における経験則に基づくものである。ピッチクラス内の個々の音の出現確率には重み付けが必要であり、さらに鳴るタイミングにも重み付けを盛り込むことは重要であるが、基本的にはスケールに基づく離散的な情報による音高の配置だけでもとりあえず「それっぽくなる」という点を重視した。

過去の自動作曲システムの失敗例の多くは、フレーズ(メロディー)を構成する前後の音の音程関係をマルコフchainなどに持込む、という点にあった。これは、せっかく「それっぽく」なったスケールなり調性(コード)感を喪失させるので、音声領域では有効でも音楽領域では適切でない、と筆者は考える。本システムにおいては、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する方針をとった。

図13は、2小節のベースパターンのうち1小節目を生成するサブパッチの例である。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、第36パラメータから第67パラメータまでの32文字でベース2小節全ての演奏情報を格納し、ドラムパートと同様に無駄のない情報圧縮を目指した。

音高方向についてはスケールを検討して、36データを「3オクターブ+5度」の範囲に割り当て、Max/MSPで内部的にこれらのパラメータをtableオブジェクトに格納した。休符(無音)は音高ゼロ、という値として定義した。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データ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。このパラメータ(tableオブジェクト内容)を表示させることで、ドラムよりも明確に、音高情報を伴う一種のメロディー情報として、ビジュアルにパターンの繰り返しや類似性を確認できる事がわかった。これは今後、グラフィクス情報からこ

のtableパターンに変換する、という新しいタイプの自動作曲(visual情報から直接にベース音楽演奏パターンを生成)の可能性としても検討していきたいと考えている。

時間軸方向については、第68パラメータの1文字の36ステップを「ベース音の生成密度」パラメータとして利用して、上記のテーブル参照を行う確率として作用させた。これにより、ベースが同じ音高のメロディー(音列)となっても、時間あたりのノート数の分布が大きくばらつくことになり、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。

ベースパートの音楽的表現としては、個々の音楽クリップごとに、ノート ON からノート OFF までの音の長さ(duration)が非常に重要である。現状ではテンポ情報に対応した比率に一定の乱数分布の幅を持つ固定値となっているが、アドバンスモードではこれをマニュアルで変更できる(生成パラメータとしては記録されないの、SMFデータ作成時の「味付け」としてのみ利用できる)ような可能性を盛り込んである。最終的に利用できる仕様とするかどうかは、本システムの重要な検討事項であると考えている。

6-7. FMC³のコード進行生成アルゴリズム

基幹となるドラムとベースのパートに加えて、本システムでは控え目な音量ながら「アルペジオパート」と「コード(バックング)パート」を加えることによって、和声(コード)の感覚を音楽クリップに持ち込んだ。「アルペジオパート」「コード(バックング)パート」の生成アルゴリズムと実装については次項で解説することとして、ここでは小節単位のコード進行の生成アルゴリズムをまず紹介する。

FMC³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であり、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。ここでは基本的にBach/Jazzの発想を継承して、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。

Bachの即興パート、あるいはJazzのインプロビゼーションにおいては、近親調や平行調への和声進行とは別に、「転調」として12平均率のどんな調にいきなり飛んでも構わない。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、「IIIm7→V7(→I)」あるいはドミナント7thコードからの半音下行(V7→Iの裏進行)があれば、そこで聴取する人間は局所的に解決(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より長期の記憶まで後付けのように再解釈させて、最終的な解決に至るテンションとして良好な印象を持つことになる。これは言い換えれば、ランダムな転調であっても局所的に広義のドミナントモーションを繰り返すことで、その音楽は刹那的には違和感なく受け入れられる事を意味する。本システムのコード進行生成アルゴリズムにおいては、この知見(仮説・理論)を最大限に活用した。

図14は、FMC³のコード進行生成パッチ部分である。ここでは、第8パラメータおよび第11パラメータから生

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、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、古典和声教科書にある近親調への和声進行、「循環コード」のようによく用いられるコード進行、PopsやRockで定番となっているコード進行などを、4小節パターンとして用意して、そこからランダムに選択することとした。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としては、以下の単純なアルゴリズムで全ての組み合わせを実装した。

ルール 1

4小節のコード進行は、「前2小節のパターン」と「後2小節のパターン」とを連結する。

ルール 2

前半の「2小節パターン」において、後ろのコードは必ずドミナント7thコードとし、前のコードは「ドミナント7th(Secondly Dominant: II7→V7)かマイナー7th(IIIm7→V7)のいずれか」とする。そしてrootは「完全5度下行か半音下行(裏5度進行)のいずれか」の2通りで進行する。これにより2*2=4通りのパターンがある。

ルール 3

4小節のうち前2小節と後2小節のパターンを連結するルールは、

- (A) 前2小節の後半のコードから後2小節の前半のコードへの隣接進行
- (B) 前2小節の後半のコードから後2小節の後半のコードへの進行(Extended Dominant)

のいずれかである。そしてrootは「完全5度下行か半音下行(裏5度進行)のいずれか」の2通りのパターンで進行する。

ルール 4

後半の「2小節パターン」において、後ろのコードは必ずドミナント7thコードとする。「O→Vm7」という進行は無いので、ルール3(A)の場合には前のコードはマイナー7thは採用されず必ずドミナント7thとする。これにより、4*2=8通りのパターンがある。

ルール 5

後半の「2小節パターン」において、ルール3(B)の場合には、前のコードも「ドミナント7thかマイナー7thのいずれか」の2通りがある。これにより、4*2*2=16通りのパターンがある。以上を組み合わせると、(A)(B)合わせて計24パターンのコード進行が生成できる。

図14にあるように、このように4小節パターンで回るコード進行は、生成された音楽クリップごとに一定であり、最初から最後まで変わらない。こ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あり、第3パラメータから第6パラメータまでの4文字(4小節単位で4回の転調設定)として、将来的な拡張の可能性を含みつつ定義した。従って生成される音楽クリップは、大きく16小節を単位として回っているのだから、転調は隣接する4小節ブロックごとの相対的な音程移動なので、16小節たっても元に戻るとは限らない(たまたま戻る確率の方がかなり低い)。16小節ごとに12音のどの音程関係に転調して