

著作権フリー音楽クリップ生成システム 「FMC³」のユーザインターフェースについて

長嶋洋一

静岡文化芸術大学

FMC³(Free Music Clip for Creative Common)とは、FLASH等のコンテンツ制作において、著作権の心配なく自由に利用できるフリー音楽クリップ、およびその自動生成システムである。本稿では2006年3月にフリー公開したFMC³のユーザインターフェース開発に関連して、幅広いユーザ層の表現要請に応え、音楽的専門知識の多寡にも対応したヒューマンインターフェースの検討について報告する。

User Interface Design for FMC³(Free Music Clip for Creative Common)

Yoichi Nagashima (*nagasm@computer.org*)

Shizuoka University of Art and Culture

This is a report of user interface design for FMC³ (Free Music Clip for Creative Common). The target user of this system is wide : designer, creator, student and amature, so I developed many levels of interfaces.

1. はじめに

2005年4月から2006年2月にかけて、「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」の開発研究として、FMC³(Free Music Clip for Creative Common)の開発を行い[1-5]、一応の完成バージョン1.02のソースプログラムやアルゴリズム解説を含む成果の全ドキュメントを2006年3月にWebサイト[6]にて公開した。

ここでは対象を「作品系マルチメディアコンテンツの音楽部分」と限定し、アルゴリズム作曲においては、楽曲生成の材料となる音楽演奏情報の素材/断片(著作物)を一切持たない自動生成とすることで、著作権の対象となる人間の介在しない自動作曲により著作権の問題の発生しないフリー音楽クリップを多数生成することができた。

本稿ではこのFMC³のユーザインターフェースの検討について紹介報告し、幅広いユーザを対象としたシステムのGUI設計について議論したい。

2. FMC³の概要

Web公開サイト[6]において、生成パラメータの仕様詳細とともに解説しているので、ここではごく簡単にFMC³のアルゴリズム作曲に

ついて整理・紹介しておく。まずバージョン1.02の全ての生成音楽クリップは、1小節を4/4とした前提の下で、1小節を

- ・8分音符8個(8beat)
- ・3連符12個(shuffle)
- ・16分音符16個(16beat)

という3種類のスタイルのいずれか、と限定した。これは1小節を48分割したタイムベースにより表現でき、該当するスタイルのビートに乗らない部分の音符情報は最初から持たないことで生成パラメータの大幅な情報圧縮を行っている。図1はメイン画面である。

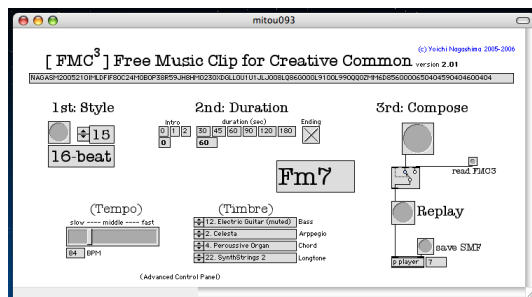


図1 FMC³のメイン画面

ドラムとベースパートについては、このタイムベース(最小限の時間軸単位)ごとに、打楽器/ベース音高の情報を生成パラメータとして持つ。バージョン1.02においてはドラム

パートは1小節だけのパターンを繰り返す。ベースパートは2小節単位のパターンを持ち、小節ごとに進行するコード上に移調してこのパターンを演奏し、上下に音域を超えた場合にはオクターブ単位に折り返す。コード進行と転調により、実際に聴取されるベースパートのフレーズは、この折り返しにより異なった(バラエティが増えた)印象を与えている。

コード進行は4小節単位で変化し、バージョン1.02では53種類のタイプからランダムに選択される。コードタイプは

- ・ドミナント7th (0-4-7-10)
- ・マイナー7th (0-3-7-10)

のたった2種類である。FMC³の特性として、個々にドミナントモーション等の連鎖として金太郎飴のように続く音楽なので、トニックが必要ない、という点が大きな特徴である。また、この2種以外のコードタイプを特徴付けるテンションノートは、他パートにより付加されるため、この枠組みのコードタイプとしては2種類で十分である。

4小節単位で決まったコード進行は、4ブロック単位でランダム設定する転調オフセット(0-11)を加算して、最終的に16小節単位の生成パターンとなる。ただし16小節で繰り返すわけではなく、次の16小節ブロックの先頭転調オフセットによりさらに別の調となる確率が11/12と大きい。一方FMC³は音楽クリップの「尺」として打切りエンディングに至る時間が絶対的に指定されるが、これはテンポによってどこまで続くかは実際に演奏生成することで確定する。従って、生成されたFMC³音楽クリップのコード進行を書き留めた場合、コードネームとしては曲全体で繰り返さなく延々と変わることがほとんどとなる。

ドラムとベースに加えて、FMC³では

- ・カッティングコード
- ・アルペジオ
- ・ロングノート(疑似メロディ)

という3つのパートを持ち、いずれもそれぞれの小節のコード(転調オフセットを付加したその瞬間のコードネーム)に対応した演奏を生成する。カッティングコードパートでは、2音ポリフォニックにより、rootや5thなどコード基幹音を除いて、そのコードタイプごとのUST(上部構成3和音)からavoid noteを除いた9th、11th、13th、さらに7th、3rdの組み合わせから選択したノートを、3種類のビートタイプごとに36種のリズムパターンのいずれかに割り当てて演奏している。

アルペジオパートでは、その小節のコードを構成する4音(root-3rd-5th-7th)を、これも36種類のパターンからランダム選択して、さらにAdvanced Modeにより設定可能な絶対的な時間差により継次生成させてアルペジオとしている。カッティングコードパートと違い、テンポとビートから与えられる、1小節48等分のタイムベースに乗らない時間的瞬間に鳴らす事でヒューマンな印象を与える。

ロングノート(疑似メロディ)パートはバージョン1の開発終盤に付加されたもので、1小節単位の2音ポリ(4分音符から全音符までの長めのシンプルなリズム)により、コードノートとテンションノートを候補として選択・付加している。同時に、8小節単位でそれぞれの小節で演奏するかどうかのON/OFFマップを持つことで、ずっと演奏し続ける不自然さを避けている。図2は、メイン画面の下に"Advanced Mode"画面が開いた例で、ここでベース・コード・アルペジオ・ロングノートの各パートの音楽的な表現に関するパラメータを設定変更できるようになっている。



図2 FMC³のAdvanced Mode画面(下)

選択的にON/OFFできる「イントロ」と「エンディング」については、それぞれ生成パラメータの36種類から、アルゴリズム生成でなく3種類のスタイルに応じて選択している。特にエンディングにおいては、楽器パートはアルペジオパートの音色を使用し、エンディング直前のコードから完全5度下行のドミナントモーションで最終コードに至り、ここに中性的な「全音音階」「sus4系」「琉球音階」などのテーブルを参照して生成することとした。コンテンツの音楽パートとしてはカットされる事も多いので、イントロ/エンディングの生成には力点を置いていない。

3. ユーザーインターフェースの検討

対象となるユーザとして、音楽的な専門知識を持たないデザイナーやクリエイターを想定したために、FMC³のユーザーインターフェースの設計においては、いわゆる既存のシーケンスソフトの複雑な画面は最初から除外した。ここではより広く、メディアコンテンツ等の制作・開発支援ツールという視点で既存のソフトウェアについて行った検討を紹介する。

3-1. ソフトウェア開発支援環境

開発(制作)支援環境として最初に整理検討したのは、図3のJava、図4のXcode(C++)のようなプログラミング環境である。ここでは、基本的にプログラミング言語(テキスト)でソースを記述するため、複数のテキストウイン

ドゥやパラメータ記述のサブウィンドウを持つ、という共通点がある。

しかしFMC³では、それ自体の開発環境としてグラフィカルなMax/MSP[7]を用いていることもあり、このような言語的な環境についてはユーザインターフェイス設計の対象から除外した。このスタイルは関数や手続きなどによる自動作曲であれば有効であるが、本プロジェクトでは対象外だからである。

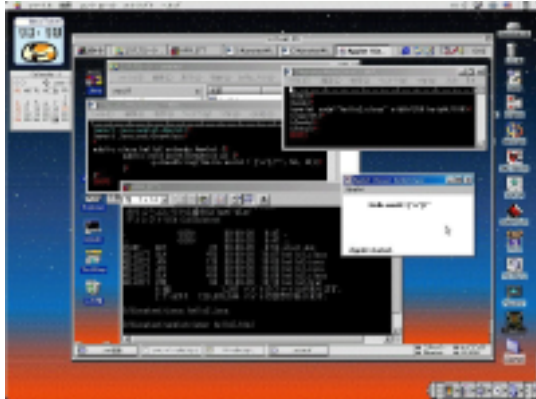


図3 Javaでのソフトウェア開発例

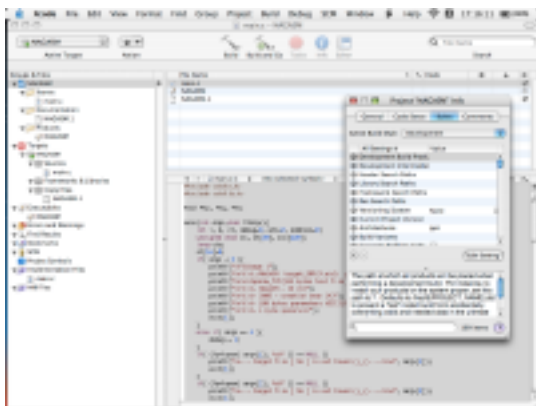


図4 Xcodeでのソフトウェア開発例

3-2. コンテンツ制作支援環境

開発(制作)支援環境として次に整理検討したのは、図5のDirector、図6のFlashのような、インタラクティブ・コンテンツの制作支援(オーサリング)ツールである。

これらのツールに共通するのは、デザイナーにCやJava等の本格的なプログラミング言語のスキルを要求しない(中級者以上では代わりにLingoやActionscriptなどの簡易言語を利用すること、タイムラインだけで記述できないコンテンツの構造を効率的に配置するための階層的なGUIにより制作を支援していることである。ただし現実には、PremiereやFinalCut等の映像編集ソフトに比べて複雑な操作方法の敷居が高く、デザイナーにとって大きな壁となっているのが実情である。この領域の制作支援ツールのGUIについては、本プロジェクトとも本質的に通じる今後の課題であると考えている。

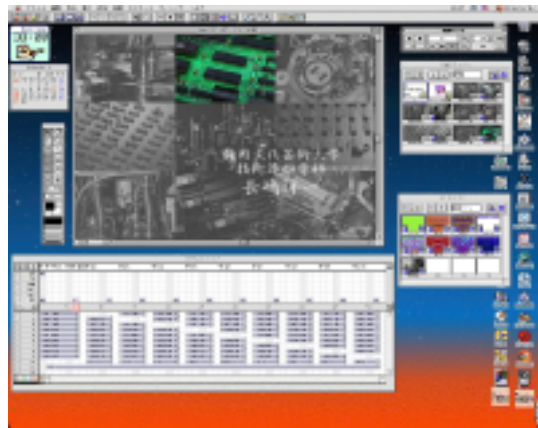


図5 Directorでのコンテンツ制作例

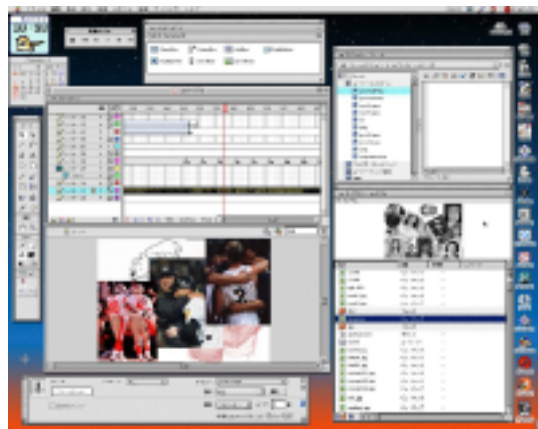


図6 Flashでのコンテンツ制作例

3-3. お手軽サウンド編集ソフトウェア

制作支援環境として次に整理検討したのは、図7のGarageBandおよび図8のSound trackという、Apple社純正の2種類の「お手軽サウンド編集ソフトウェア」である。



図7 GarageBandの画面例

図7のGarageBandはMacOSXに標準添付されているサウンド編集ソフトで、同様に標準添付されている映像編集ソフトiMovieや音楽再生ソフトiTunesなどと有機的に連携することで、オリジナルのムービー作品などを手軽に制作できる、というのがコンセプトとなっている。その基本は「多数のサウンド素材を好

きに呼び出し並べて繋げる」というものであり、複雑高度なMIDIシーケンスソフトより親しみ易いGUIであるものの、基本的な思想はタイムラインに素材を配置する、という古典的な制作環境となっている。



図8 Soundtrackの画面例

図8のSoundtrackは、Appleの映像編集ソフトFinalCutとバンドルされているサウンド編集ソフトで、機能も対象もアドバンスとなっているが、ここまで来ると、本格的な音楽編集ソフトの代表であるProTools(図9)とほとんど変わらないGUIとなり、音楽を専門としない一般のデザイナー/クリエイターにとって、得られるサウンドの魅力は大きいものの、難易度は格段にアップしてしまう。そこで本プロジェクトのインターフェース設計では、この路線は追求しない事とした。

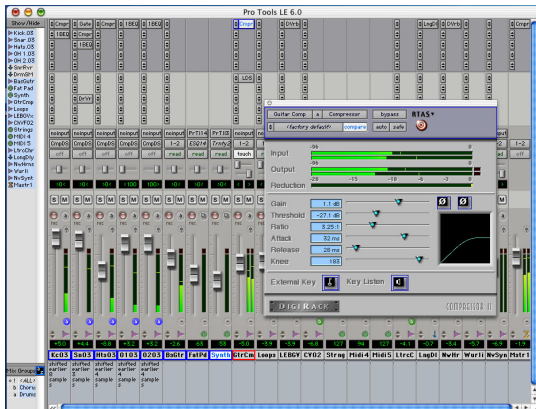
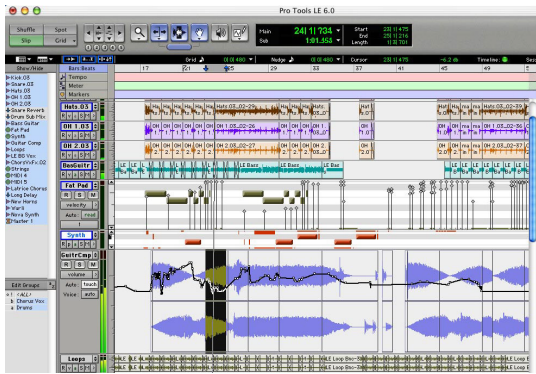


図9 ProToolsの画面例

3-4. お手軽ムービー制作ソフトウェア

FMC³のユーザーインターフェースの検討において重要な参考となったのは音楽生成ソフトではなく、図10のLiFEwithPhotoCinema (DigitalStage社)である。このソフトは、携帯やデジカメなどで多数のJPEG写真を用意し、さらにMP3形式のサウンドファイルを用意すると、最短の場合には「おまかせ」ボタンによって、そのMP3ファイルの演奏時間の長さを持ち、多数のJPEG静止画を多種のエフェクトアニメーションで操作したムービーをたった数秒で自動生成してしまふ、という画期的なシステムである。



図10 LiFEwithPhotoCinemaの画面例

著作権の点で問題となるのはこのMP3サウンドファイルで、自分のお気に入りの市販CD等の楽曲を使えば、これはprivateにしか楽しめなくなる。FMC³はまさにこの部分に対応したものであり、実際に公開サイト[6]には、FMC³でサウンドトラックを自動生成し、ムービーをLiFEwithPhotoCinemaで「お任せ」作成したサンプルムービー(Shockwave形式、MPEG4形式)が並んでいる。

3-5. お手軽BGM制作ソフトウェア

FMC³の自動作曲アルゴリズムの検討において参考としたのは、この分野の老舗であるBand-in-a-Box(図11)である。

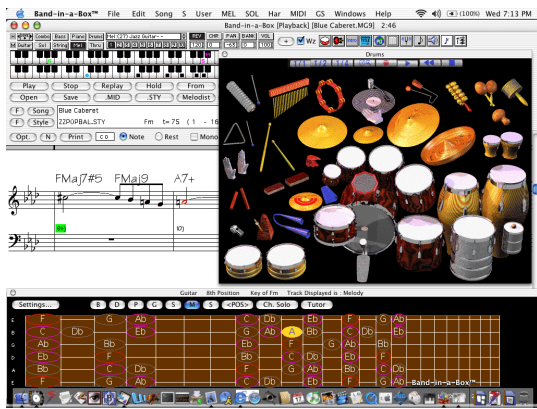


図11 Band-in-a-Boxの画面例

このソフトウェアは、コード進行をシステムの基幹に据えて、多種のスタイル、メロディの自動生成などを広範に支援するものであるが、やはり音楽的な基礎がないユーザにとっては使い易いとは言えない。本プロジェクトにおいては、色々な音楽的機能についてBand-in-a-Boxを参考として検討した部分もあり、本稿では次に、この各機能に対応したユーザインターフェースの検討に関して、対応するBand-in-a-Boxの編集画面を比較して紹介することとした。

4. 音楽的機能ごとのGUIの検討

4-1. スタイルの選択

FMC³を公開サイト[6]からダウンロードして解凍したユーザは、図1のメイン画面において、以下のような操作を行う。

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」と「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) 演奏開始

これだけである。当然ながら、このオペレーションに音楽的知識はまったく要求されない。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、試聴後に希望すれば「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、簡単にAIFFやMP3に変換できる。

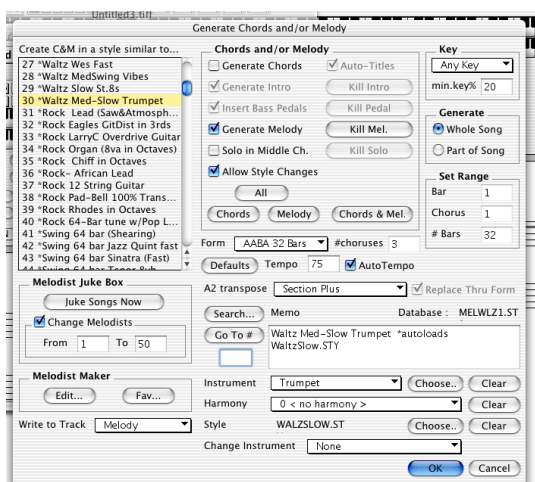


図12 Band-in-a-Boxのスタイル生成例

一方、Band-in-a-Boxでは、図12のようなテキストベースのユーザインターフェースによって、コードやメロディのスタイルを指定して自動生成する。ここでは、音楽的な知識とともに英語力も必要となる。音楽的知識についても、コードネームや楽典に載っているような教科書的知識と、「アフリカ風」「シ

ナトラ風」というような蘊蓄系知識とが混在しており、初心者の困惑を増大させている。そこでFMC³ではこの部分はユーザから伏せて内部で生成する方針とした。

4-2. 全体構成とコードの選択

FMC³の第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、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。一方、Band-in-a-Boxでは、図13の上段のような「コード進行の時間的配置」により音楽の全体構成を制作し、下段のようなメニューにより具体的なコードネームを設定する。前述のように、FMC³にはたった2種類のコードタイプしかなく、刻々と転調を繰り返す音楽的構造のため、このような設定画面は不要である。

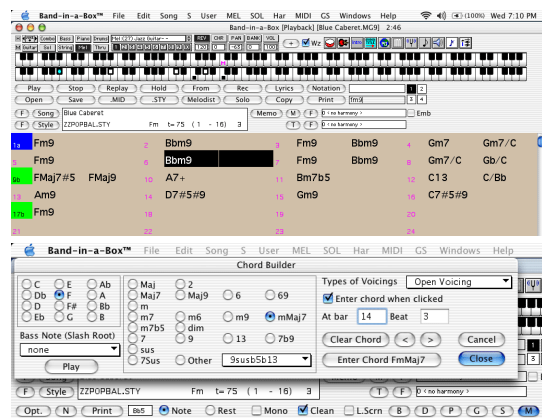


図13 Band-in-a-Boxの設定画面例

4-3. 100文字パラメータの隠蔽と公開

FMC³の自動作曲ステップでは、36進数100文字パラメータを図14のようなパッチで生成している。このテーブルをユーザに公開することは容易であるが、実装において不可視と指定し、編集できない仕様とした。

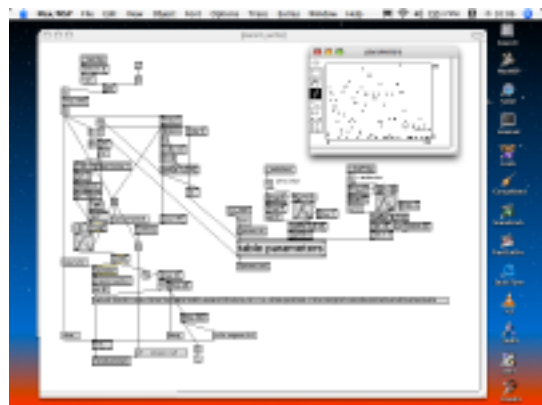


図14 FMC³の内部パラメータ部分

これはFMC³の基本的な設計指針でもある。「音楽の専門知識の必要なくいくらかでも音楽クリップを生成する」という思想に基づくものである。Computer Musicのアルゴリズム作曲の領域の視点からは、図14のパラメータテーブルをユーザに公開することで、新しい対話的自動作曲システムの可能性を提供できるのは分かっている上での判断である。

4-4. ドラム/ベースの隠蔽と公開

FMC³の自動作曲ステップでは、1小節単位のループで回るドラムパートに、楽器としてバスドラム(ON/OFF)・スネア(ON/OFF)・Hi-Hat(open/closed/OFF)・clap/ride/OFF、の6種を割り当て、これらをタイミングごとに2*2*3*3=36通りの1文字にエンコードしている。タイミングも48ポイントから圧縮して第76パラメータから第99パラメータまでの24文字で全て格納した。ベースは2小節パターンのループであり、時間軸方向では最大16ステップに圧縮、音高方向についても36データを「3オクターブ+5度」の範囲に割り当て、第36パラメータから第67パラメータまでの32文字で全てを格納した。

これらのデータはそのまま一種のドラムマシン、シーケンスデータとなっており、FMC³の内部パッチの開発段階においても、図15のように実際に個別にON/OFF変更やノートの変更ができるGUIを実装している。当初はこの画面をユーザがopenできるように想定していたが、前述のようなインターフェースの検討の中で、最終的にはこのサブパッチ自身をバージョン1.02においては不可視設定として、ユーザはこのドラムとベースのパターンを個別に編集できなくする、と決定した。

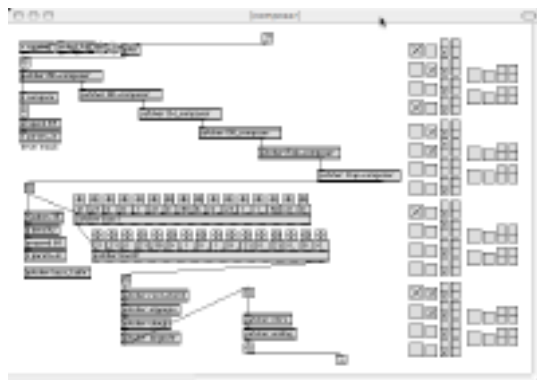


図15 FMC³のドラム/ベース設定部分

5. 検討と考察

基本に立ち返ってみれば、FMC³の目標は背景音楽(BGM)パートをお手軽に[使える]音楽データとして自動生成することである。これは短期的にはクリエイター(とその卵)のためのシステムであるが、2年後に到来する「クリエイション(作品創造)の大衆化」という時

代的要請を視野に入れている。このユーザに、専門的/高度なMIDIシーケンサの機能やパラメータ、音楽理論や楽典の知識、音源やアンサンブルの知識などを前提とするのは優しくないインターフェースであろう、との視点から、「お任せ」をキーワードとして図1のようなシンプルなメイン画面とした。

最後まで悩んだ図15の個別編集機能については、このパッチを可視化するとさらに内部構造が見えてしまうこともあり、default機能として伏せるように決定した。一方、公開サイト[6]ではFMC³のソースを詳細な解説とともに全て公開しており、希望すれば全てを解析・改造できる体制である。現実には、ネットでFMC³を知った一部のユーザは、音楽情報科学研究のためにソースの検討を行うと知らせてきた。もしFMC³に絢爛豪華・詳細なユーザインターフェースを実装したいのであれば、それこそ「お任せ」である。

本プロジェクトに限らない課題として、今後、メディアコンテンツのオーサリングツールのユーザインターフェースについても検討する計画である。

6. おわりに

「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」FMC³のユーザインターフェースの検討について紹介報告し、幅広いユーザを対象としたシステムのGUI設計について検討した。今後、単独で聴取できる音楽を目指したメロディ生成アルゴリズムの検討などのテーマとともに、新しいバージョンに向けて、さらに検討改良を進めていきたい。

参考文献

- [1] 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
- [2] 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
- [3] 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
- [4] 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平成17年度秋季研究発表会資料, 日本音楽知覚認知学会, 2005
- [5] 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
- [6] <http://suac.net/FMC3/>
- [7] <http://www.cycling74.com/>