

Twitter を用いた多人数参加型作曲システム

吉田周平[†] 星野厚^{††} 浜中雅俊^{†††}

本論文では Twitter への単旋律データの投稿と、これらを素材として再利用することで作曲を行うことのできるシステムについて述べる。素材の再利用によって作られた曲データは再び Twitter に投稿でき、また改編可能にすることで未完成状態での投稿や、作曲の過程を共有することができる。

多くの作曲システムは一人で作曲することが前提のシステムとなっており、素材データの作成、共有という概念はなく、プロの作った素材を購入して使用するのが一般的となっていた。

そこで我々は Twitter のつぶやきのような気軽さで旋律を投稿でき、ユーザー同士のつながりを生かした多人数参加型の作曲システムを設計することを考えた。具体的にはデータ形式として MML (Music Macro Language) を採用することで、テキストで音楽を表現することを可能とした。そして素材の作成から Twitter への投稿、投稿された素材を利用した作曲とその曲の Twitter への投稿、さらに投稿された曲の改変までを統合的に行えるシステムを実装した。

Twitter based composing system for remote people

Shuhei Yoshida[†] Atsushi Hoshino^{††} and Masatoshi Hamanaka^{†††}

This paper describes a composing system, which can post a monophonic melody to Twitter and can create a song by the placement of the posted melodies as materials to Twitter. The post of an unfinished song and sharing the compositional process by allowing modifying. Most previous composing system were for single user and there were no concept of creating and sharing materials, and generally used to purchase a materials produced by professional. So we thought to design a composing system for remote people which can post a melody as simple as tweet at Twitter and use connections between users. We implemented a system using MML (Music Macro Language) data format which enables write a music in text format. We implemented the system which record a melody, post the melody to Twitter, create a music by using posted melodies, post the music to Twitter, and modify the posted music.

1. はじめに

本研究では、Twitter への数小節の楽曲データの投稿と、それらを楽曲のパーツとして再利用し作曲を行った結果を再度投稿するシステムを設計する。本研究の特長は他のユーザーによって投稿された曲に使用されている素材の配置情報を確認、改変が可能なことであり、改変が可能なことによって楽曲データが再利用可能な形で公開され、既存の曲に対し自分でソロを追加することやパート別の作曲、著名作曲家との仮想的な共演などが可能となる。本システムでは数小節程度の演奏や音色の作成、パーツの配置提案等のみで楽曲制作に関わる事が可能になる。多くの作曲システムでは一人で作曲を行う事が前提とされてきたが、本システムによって作曲に対する敷居を下げられる事も魅力である。

従来の作曲ソフトの例として ACID Pro[1]や GarageBand[2]といったループベース DAW (Digital Audio Workstation) と呼ばれるソフトウェアが挙げられる。ループベース DAW は数小節の楽曲素材 (ループと呼ばれる) をタイムラインに配置することで作曲を行うので、初心者でも比較的簡単に作曲することが可能である。しかし個人での使用が前提であり、また素材はソフトにプリセットとして付属するものを使用し、またプロの作った素材 CD を購入するのが一般的な素材の追加方法であった。

一方音楽のネットワーク共有に関する研究例では遠隔地間の合奏やセッションなどリアルタイムでの音楽の共有が提案されていたが[3,4]、作曲過程を共有する機能はなかった。

既存のシステムで作曲過程の共有が行われてこなかったことに対し、我々は情報が人から人へ渡り、その過程でまた新たな情報が付加されていくという特徴を持つ Twitter に着目し、これを作曲過程の共有に利用することが可能ではないかと考えた。Twitter は最近注目を集めるミニブログサービスであり、学習情報の共有化に関する研究[5]での利用例や、Twitter を用いたオンラインホワイトボードサービス[6]など様々な分野において「情報の共有」をキーワードとして利用され始めている。また Twitter は著名人とでも簡単に繋がりを持ち、同じ目線でネットワークに参加することが可能である。

本研究では Twitter のつぶやきのような気軽さで音楽を録音、編集、投稿するシステムの構築を目指すにあたり、音楽をそのまま Twitter に投稿するためにテキストで楽譜情報を記述する MML (Music Macro Language) を採用することで Twitter の利点を楽曲素材の共有に生かすことを可能とする。そして気軽に投稿された旋律を楽曲の素材として再利用すること、さらに気軽に投稿された旋律を再利用することによって完成した曲の改編を許容することにより作業の分担が可能となり、一人で作曲することの敷居の高さを克服するようなシステムを構築することで、Twitter でのテキスト形式のつぶやき同様に作曲作業が人から人に渡っていくという新しい音楽のスタイルや文化を生むことが可能ではないかと考えた。

そこで我々は鍵盤によって旋律素材を録音し、Twitter に投稿されたお気に入りの素材を配置することで作曲が行え、作成した素材と曲データを Twitter 上でやり取りすることが可能なシステムを iPhone アプリケーションとして実装した。

2. システムの概要

本システムは Twitter を介し楽曲制作を共同で行う事が可能な iPhone アプリケーションである。システムの概略図を図 1 に示す。

素材と楽曲データの入力、投稿、閲覧、配置編集機能はすべて単一の iPhone アプリケーションに実装する。ユーザーは iPhone に表示された鍵盤から旋律を録音し、録音したデータを素材として Twitter に投稿する。投稿された素材は他のユーザーから視聴することが可能である。各ユーザーは投稿された素材の中から好みのものをお気に入りとして保存し、保存した素材を再利用し旋律を重ねたり前後に並べるなどして一つの曲を完成させる。本システムではそれぞれのデータについて以下のように名称を付け、定義した。

- ・**ブロック** : 鍵盤によって録音された、曲の素材となる単旋律のデータ
- ・**ソング** : ブロックが集まり、配置されることで作曲された曲データ

作ったソングは再度 Twitter に投稿することが可能である。各ユーザーは投稿されたソングを保存することで、メロディやソロの追加といった改変を加えられる。また各ソングには元になったブロックの URL が保存され、その作者を辿ることや、その作者の別のブロックを探すことも可能である。従ってソングの改変によりユーザー同士の繋がりが連鎖していく。

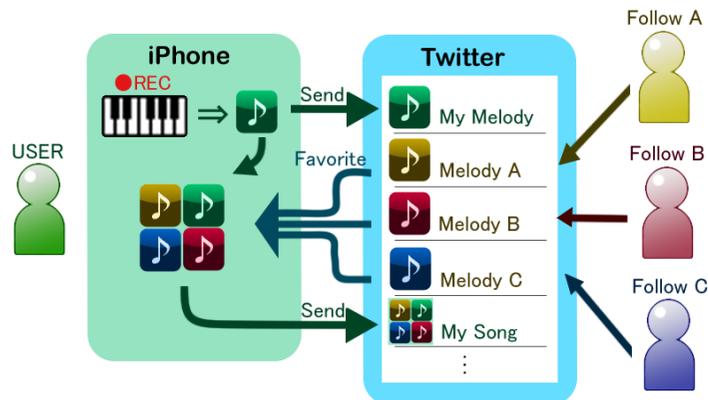


図 1 システムの概略図

2.1 アプリケーションの構造

この節ではアプリケーションで行う構造について説明する。図 2 は各画面をベースにしたアプリケーションの構造を表している。鍵盤で録音したブロック、ブロックの編集で作ったソングはリストに保存が可能であり、リストから Twitter に投稿するデータを選択する。Twitter からお気に入り登録したブロック、ソングは別のリストに保存され、保存したブロックを用いてソングを作成したり、ソングを編集する、といった流れである。各画面はメニュー画面から移動する。3 章以降で各画面における機能について詳しく説明する。

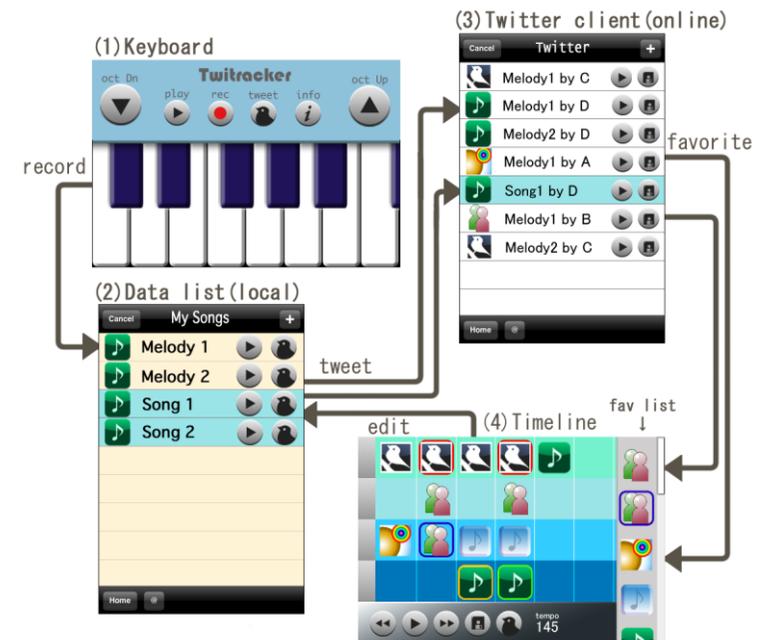


図 2 アプリケーションの構造

2.2 楽譜記述言語 MML について

MML は MIDI(Musical Instrument Digital Interface)規格が浸透する以前にコンピューターミュージックにおいて主に使用されていた楽譜記述言語であり、テキストで楽譜を表現することが可能である。元々N88BASIC[7]などで PC の内蔵音源を使用するた

めの言語であったが、MML は記述しやすさから簡易の楽譜記述言語として使われ続け、現在でもシーケンスソフトウェア等に採用されている例がある。本研究でも MML を楽譜記述言語として扱う。インターフェイス上では MML を意識することなく録音、編集を行えるような設計を目指す。MML に慣れたユーザーであれば MML レベルでの編集も可能となる。

MML の記述方法は至ってシンプルであり、音符の表現は音階の英語表記をそのまま記述すればよい。例えば「ドレミファソ」は以下のように記述する。

CDEFG;

多くの場合初期設定では一文字の発音長はテンポ 120 の 4 分音符であり、これを変更しない場合は上の例の記述だけですぐに再生することが可能である。

またテンポやオクターブ、半音など鍵盤からの録音に必須となるパラメータは基本的に備わっており、音源の操作によって音色の変更が可能であるため、本研究のブロックのような簡単な演奏情報を記録するには十分である。

2.3 音源の実装

MML で記述されたデータの再生には MML に対応した音源が必要になる。MML の再生が可能な音源のひとつに、flash の ActionScript によって作られた FIMML[8]というものがある。FIMML はオープンソースで、環境によってパラメータ設定の表現が異なる事が多い（方言と呼ばれる）MML において基本的な表現で制作が可能であり、機能も豊富である。また URL 内に MML を記述することで引数に取り、リンクのクリックですぐに視聴が可能となる FIMML プレイヤー機能なども実装されている。このため、

- ・ピコカキコ[9]（ニコニコ動画内に設置された MML リアルタイム視聴、編集の可能な掲示板）

- ・kuchibue[10]（MML を投稿すると FIMML プレイヤーへのリンクが生成される Twitter サービス）

など様々なサービスに利用されている。

本システムは iPhone アプリケーションとして実装するため、flash を使用することが困難なため iPhone 用の音源を新規にアプリケーション内に実装する。また FIMML への準拠を目標とする。

3. 録音から投稿まで

素材の録音と投稿は一つの画面で行う事を可能にする。録音と投稿は本システムで作曲を行う上で基礎となる機能であるため、誰にとっても分かりやすく、また思いつ

いた時にすぐ投稿できるような見通しの良いものにする必要がある。この章ではその設計と仕様について説明する。

3.1 録音

録音時の画面を図 3 に示す。ユーザーが一目で使い方を想像できるインターフェイスを目指し、概観は一般的な鍵盤とする。またテープレコーダーのような操作感の録音ボタン（図 3 上部）を用い、MML 等内部処理を意識せずに録音することを可能にする。ボタンを押すとメトロノーム音とともに一小節前からのカウントダウンが始まり、カウントダウン以降の演奏結果が録音される。2, 4, 8 小節などの中からユーザーが選択した指定小節数に到達すると録音は自動的停止し、今の演奏結果を保存するかどうかを問うダイアログが表示される。また、tweet ボタンを押すことで演奏結果をすぐに投稿でき、画面を切り替えることなく録音から投稿までを行う事ができる。

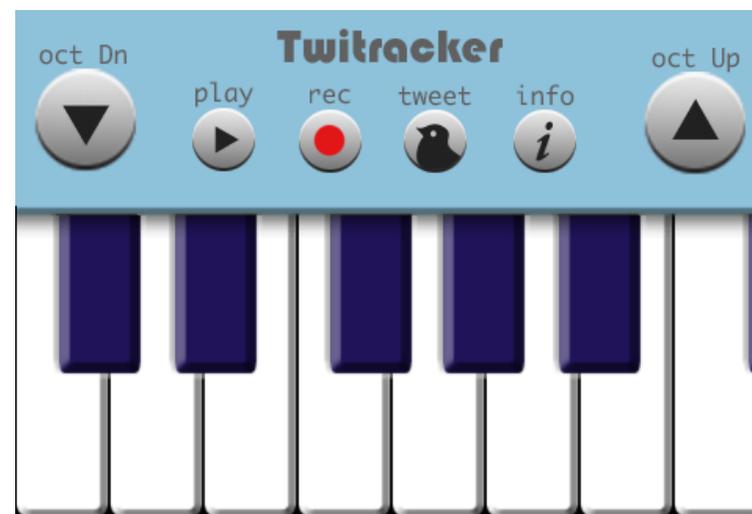


図 3 鍵盤インターフェイス

録音によって保存されるデータは MML であり、旋律やテンポなどの情報を含む。ここで録音できるのは単旋律のみである。これをブロックと呼び、これを単位として編集を行っていく（5 章で説明する）。この編集において、ブロックの配置により旋律を重ねることで和音を実現する事ができる。

また、この録音作業によってブロックは iPhone 上に保存される。ブロックのリストは

アプリケーションから確認でき、その画面からも視聴と投稿を行うことが可能である (図 4)。



図 4 データリスト画面

3.2 Twitter への投稿

作成したブロックとソングは Twitter に MML, つまりテキストとして投稿する。しかし Twitter には投稿可能な文字数が 140 字までという制限がある。この中に MML を直接記述する場合、表現が制限されてしまう。例えば一文字を 4 分音符とすると、単純に計算して単旋律で 35 小節分となるが、ソングの投稿の場合 4 和音で約 9 小節分しか記述できない。また使用ブロック作者の URL を含めたり、8 分、16 分音符の使用によって更に文字数が増えるため、この中で MML をそのまま表現するには不足であり、

なんらかの対策が必要となる。

文字数の制限に対し、本システムでは送信時にデータを URL 内に記述する方法をとり、更に URL 短縮サービスを利用することで文字数の制限の問題を解決する。

URL 短縮サービスでは数文字の文字列を生成し、これを ID とすることで元の URL を呼び出す方式を取っているため、ソングデータに含むべき使用ブロックの URL は URL 短縮サービスにおける ID となる文字列 (多くの場合 4~6 文字) だけで表現する事が可能である。また URL で投稿する利点として、リンク先を FIMML プレイヤーとすることで、flash 対応デバイスからアクセスした場合は投稿データが FIMML プレイヤーに送られ、すぐに視聴することが可能である。

4. 投稿されたデータの閲覧と再利用

様々なユーザーから投稿されたブロック、ソングを iPhone で閲覧するために、アプリケーション内に Twitter クライアント機能を実装する。各投稿には視聴ボタンとお気に入り登録ボタンが表示される。視聴ボタンからはその場で視聴が行え、お気に入りボタンからは好みのブロック、ソングをローカルに保存する事が可能である (図 5)。演奏中、再生ボタンは停止ボタンに切り替わる。

お気に入りに登録し保存されたブロックとソングは、5 章で説明するソング編集に使用することが可能である。つまりブロックはソングを構成するパーツとして使用し、ソングは再編集し改変を加えることが可能である。

5. ソング編集と投稿について

この章ではブロックの配置によって曲を作るソング編集について説明する。

5.1 概要と操作

図 6 はソングの編集画面である。画面右側にお気に入りに登録したブロック一覧のリスト、中央に各パートのタイムラインが表示される。ブロック一覧は投稿者の Twitter アイコンとして表示され、リストに表示されたアイコンをタイムラインにドラッグ&ドロップすることで配置する (同一投稿者のブロックが複数ある場合は枠線が付き、その色により区別する)。タイムラインでは縦軸はトラック (パート)、横軸は時間を表している。

ソング編集画面で行えるのはブロックの配置と全体テンポの設定、各ブロックのトランスポーズ (転調) 設定であり、単体のブロックに対しては保護のためトランスポーズ以外の変更 (例えば音符単位での編集など) は加えることは禁止する。

各ブロックの視聴はリスト内並びに配置されたブロックをタッチすることで行う。トランスポーズ設定は配置済みのブロックを上下にスライドすることで行う。

また画面下のボタンから再生や再生ヘッド位置の移動、ソングの保存と投稿をそれ

ぞれ行う事が可能である。

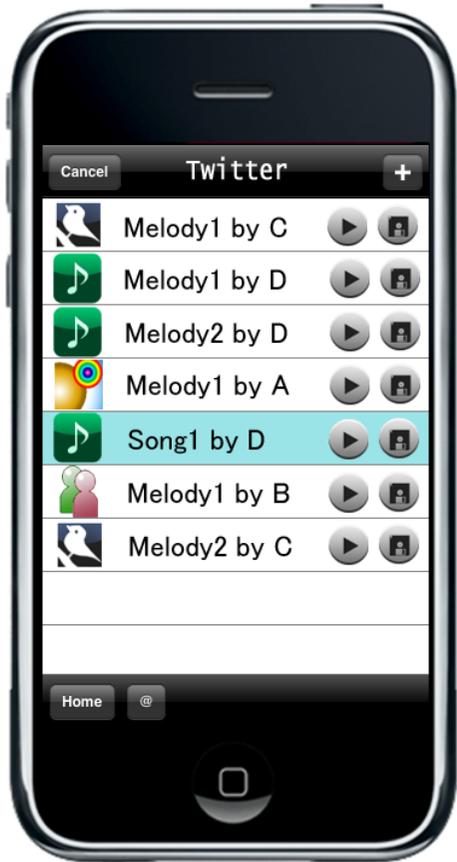


図 5 Twitter クライアント画面

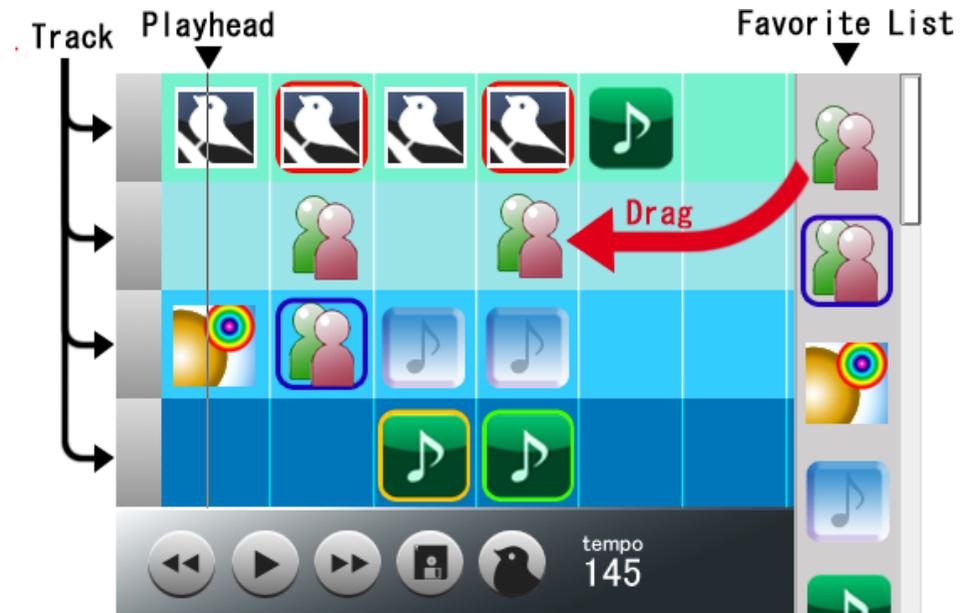


図 6 ソング編集画面

5.2 実現方法

ソング中に用いられるブロックは、その URL から MML に展開し、ブロックの配置と組み替えは、本システムで MML の再生において準拠を目標とする音源である FIMML の置き換えマクロ機能を利用する。

FIMML の置き換えマクロ機能とは、曲中に繰り返される部分がある時、何度も同じ文字列を記述する必要が無いようにその部分を短い文字列で定義することを可能にするものであり「\$定義名=内容」という書式で記述が可能である。以下に使用例を示す。

\$A = CDE;

\$A R \$A R GED \$A DE;

ソング中に用いられたブロックは MML で置き換えマクロとして定義し、作業は置き換えマクロによる置換文字列を使用して行う。またブロックの URL をコメントとして併記することで元データへのリンクを保持する。ソング投稿時にはこのリンクから使用ブロックの作成者情報などを付加して投稿する事が可能である。

5.3 投稿方法

ソングの編集結果もブロックと同様に iPhone 上に保存することが可能であり、そのリストから視聴と投稿を行う (図 4)。

投稿されるソングに含まれるデータは、

- ・このデータがソングであることを示すフラグ
- ・使用したブロックの MML、URL
- ・テンポや拍子等の全体パラメータ
- ・ブロックの配置位置情報

である。ソングの投稿時にはデータを URL とし再度 URL 短縮サービスにかけることで、投稿文字数はブロックと同程度となる。Twitter クライアントからの閲覧時にはヘッダによりブロックと区別され、色分けされて表示される。

6. まとめと今後の課題

本研究では楽曲の素材となるブロック、ブロックを使って作られた曲データであるソングを簡単に作成し、Twitter 上で公開、共有するための iPhone アプリケーションを制作した。これによって素材と作曲過程の共有が容易になり、初心者でも簡単に制作に参加することが可能になった。

実装に用いたデバイスについて、PC の場合は入力機器の中でも鍵盤 (MIDI キーボード) は一般的ではない事、また携帯に不便なため、今回はそのものを入力機器として利用できるタッチ方式のスマートフォン、中でもユーザーに親しまれている iPhone を利用してシステムを設計した。しかし本システムの考え方自体はデバイスによらないため、様々なプラットフォームに移植していく事を考えている。

今後システムを公開しユーザーの使用状況をみてシステムの評価方法を検討していく。また投稿された楽曲が著作物であった場合に対する対策なども行っていく予定である。

参考文献

- 1) SONY 「ACID Pro 7」 <http://www.sonycreativesoftware.com/acidpro> (2008)
- 2) Apple 「GarageBand」 <http://www.apple.com/ilife/garageband/> (2009)
- 3) 後藤 真孝, 根山 亮, 菊地 淑晃, 村岡 洋一 「RMCP: Remote Media Control Protocol: 時間管理機能の拡張と遅延を考慮した遠隔地間の合奏」 Information Processing Society of Japan (IPSI) SIG Notes MUS-021 pp.13-20 (1997)
- 4) 長嶋 洋一 「GDS(global delayed session) Music --- ネットワーク遅延を伴う音楽セッション・モデル」 情報処理学会研究報告[音楽情報科学] MUS-046 pp.61-66 (2002)
- 5) 古性 淑子 「Twitter サービスを利用した学習情報共有化について」 横浜美術短期大学教育・研究紀要 (2010)
- 6) 「99 Basic と N88 互換 BASIC への手引書」 <http://homepage3.nifty.com/skis/n88/index.htm> (2006)
- 7) 「FlockDraw」 <http://flockdraw.com/> (2009)
- 8) 「FIMML」 <http://coderepos.org/share/wiki/FIMML> (2009)
- 9) 「ピコカキコ」
<http://dic.nicovideo.jp/a/%E3%83%94%E3%82%B3%E3%82%AB%E3%82%AD%E3%82%B3> (2008)
- 10) 「kuchibue」 <http://twitter.com/kuchibue> (2009)