

リアルタイムで共同作曲ができるWebアプリの開発

藤澤 伸[†] 服部 隆志[†]

慶應義塾大学環境情報学部[†]

1 はじめに

この論文のテーマは、共同作曲のツールとしてのWebアプリケーションの開発である。技術や設備等が無い全ての人が、作曲及び共同作曲に参加できるような環境を作ることが目的である。またその過程で、音楽のリアルタイムな共同編集の手法についても言及する。

2 作曲における問題点

作曲という言葉をここでは、メロディを発想し、記録し、更に別のメロディ(パート)と組み合わせ、曲としてのまとまりがある状態まで完成させること、を指すものとする。

人が作曲を始めようと思った際に問題となるのは、音楽理論の知識や経験、記録・再生をするための媒体が必要だということだろう。音楽に関する知識が無ければ、そのメロディの調性や妥当性を検証することはできず、別のメロディを組み合わせたりすることも困難だろう。また記録・再生をする媒体が無ければ、作曲は困難になる。そうした音楽を記録・再生するための媒体の多くは導入コストが高いため、やはり作曲を始めようとする人の妨げになっている。

本論では、これらの問題を解決するツールとして、共同作曲ができるWebアプリケーションを提案・開発する。音楽の知識が未熟なユーザーでも、共同作曲であれば、知識が豊富なユーザーの助けを借りながら、曲をまとめることができる。またWebアプリケーションならば、インターネットに接続できるコンピューターとWebブラウザが揃っていれば利用可能であり、記録・再生のための媒体を準備する手間やコストは大きく削減される。

3 先行研究との比較

情報技術を用いた共同作曲は、単純に作曲経験者たちの関心を引くものであり、そのためのWebサービスも既に存在する。例えば「Yourself Music (<http://yourselfmusic.jp/>)」という、音楽共同編集コミュニティサイトがある。このWebサービスでは、ユーザーがピアノロールインタフェース(5線譜が読めなくても利用可能)で曲を入力し、保存・共有ができ、他のユーザーもそれを再生したり、その曲を元にした別の曲を作成することもできる。しかしながらこの方法では、共同作曲を行う上での問題も残る。まず、曲は1度完成した状態にしなければ、公開・共有できないという点だ。すなわち、ひとりでそのレベルまで曲を仕上げる音楽知識が無いユーザーは、共同作曲まで至ることができないのだ。また、他のユーザーが投稿した曲を編集して、自分の曲として保存することはできるが、同時進行でひとつの曲を作るような形は想定されていない。つまり他のユーザーの意見を取り入れるには、1度曲を保存し、手を加える際にはまた別の曲として保存しなくてはならないため、前章で述べたような、他のユーザーの知識を借りながら進めていくような共同作曲には向いていないと言える。

4 実装

前章で述べた「Yourself Music」で見られたような問題を回避するために、リアルタイムでの共同作曲を前提としたアプリケーションを開発ことにした。すなわち、複数のユーザーが同時に曲へログインし、保存も逐次行われる。これにより、曲を完成状態まで作る前から共有が行われるようになる。また、保存先も全員で共有される。

リアルタイムな共同編集で問題になるのは、複数のユーザーが同時にデータの変更を加えた時に、どのように統合するかだ。例えば、あるユーザーAが

未保存の状態、あるメロディを編集しているときに、別のユーザーBが同じメロディに変更を加えて保存した場合、すぐにその変更を反映するとユーザーAの編集は破棄されてしまう。

音楽以外の場合は、全ての変更をバージョンとして保持し、後に最適なバージョンに状態を戻したり、複数のバージョンをマージする、という方法もある。しかし音楽の場合は、バージョンを複数保持していても、比較するにはひとつひとつ再生するほか無く、かなりの時間がかかる。また変更のそれぞれには、コメントなどでは表し切れない微妙な意図が含まれる場合が多く、機械的にマージすることは難しいと考えられる。そのため、バージョン方式で変更を管理するのは向いていないと言える。

そのため本アプリケーションでは、ある曲の作曲に参加しているユーザーそれぞれに「パート」を与え、他の「パート」を編集できないようにした。これにより、変更の衝突を防ぎ、ユーザーごとに担当する音域や章を分け、かつお互いの作業に影響を与え合うことができる。編集は、パートを選択して、マウスクリックで1音ずつ音程と時間軸上の位置を指定する。



なお編集中は自分の「パート」のみを再生するモードと、他のユーザーの「パート」も全て再生するモードを1クリックで切り替えられるようになっている。

なお実装では、言語としてJavascriptとRubyを用いた。音声の再生にはHTML5のAudio要素を用い、保存時などの通信はすべてAjaxを用い、ページ遷移で作業を中断させることなく、他のユーザーによる変更も素早く反映される。

5 実験

まずは以下のような手順で、3名の被験者に実際にアプリケーションを試用してもらい、その体験について質問をした。

- (1) 本アプリケーションを使って、まずは被験者ひとりでメロディ作成を行ってもらおう。操作方法などの質問にはその都度回答した。(5分程度)
- (2) 続いて、開発者も編集集中の楽曲に参加し、共同で楽曲制作を行う。(10分程度)
- (3) アンケートに回答させる。(5分程度)

被験者は、操作説明が済んだ後は、問題無く操作を行っており、またアンケートの「『共同作曲』ができたと思いませんか?」という問いには全員が「はい」と答えた。故に、本アプリケーションは共同作曲のツールとして利用できるものだと考える。

またアンケートの「お互いの言葉でなく、音の入力でコミュニケーションがとれましたか?」という問いには、被験者の全員が「はい」と回答した。そのうち「音の入力でコミュニケーションが行われた時、その影響で自分のパートを修正しましたか?」という問いにも全員が「はい」と答え、本アプリケーションでのコミュニケーションが、メロディの編集を促し、影響を与えていることが予想される。まだ被験者も少なく、評価には至らないが、コミュニケーションが発生しやすい傾向にあると言える。

6 結び

実験より、本アプリケーションが音楽の記録・再生媒体として使えること、共同作曲を可能にし、作曲を支援するコミュニケーションの場となることが期待される。したがって本稿序盤で述べた、技術や設備等が無い全ての人が、共同作曲に参加できるような環境を作るという目標に、貢献できるのではないかと考える。今後被験者を増やし、他のツールとの比較も行い、より厳密な評価を行いたい。

7 参考文献

- [1] 木村 昌樹 「リアルタイム協調作曲のためのコミュニケーション支援システム」, 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, 2010