5Z－1

軽音楽におけるアドリブ演奏の計算機による自動生成プログラムの実装

岸田良朗　　林恒俊
立命館大学　理工学部

1　はじめに

音楽情報処理においては，自動作曲等，様々なテーマについて研究がなされ，多くの成果が得られてきている [1] [2]．本研究は，軽音楽における即興演奏を計算機を用いて自動生成することを目的としている．軽音楽におけるアドリブ演奏は，芸術的な価値判断の基準についての自由度が高いため，プログラムの仕様自体の自由度も高くなると考えられる．これまでに，実際の演奏者の知識を蓄積し，そのデータを基に，伴奏情報を参照しながらアドリブを自動生成するプログラム Boogie Project(以下 BoP 1) の開発と実装を行って来た [3]．

プロジェクトは，新しく Boogie Project 2(以下 BoP 2) と名付けられ，MIDI プロトコルの取り扱いとデータの処理が実装され，自動生成されたアドリブはリアルタイムにオーディオ装置から出力されるようになった．また，自動生成プログラムをより高度に発展させ，より自然で音楽的な評価が得られるような演奏を生成することができるようにした．

2　BoP 2 の仕様

BoP 1 同様，実際の演奏者 (プレーヤー) のアドリブを構成する小さな単位となるアドリブ・フレーズをドキュメント・ファイルに収集する．各アドリブ・フレーズには，コードとの相性，音楽的な緊張感の程度などの属性を，小節の中での開始位置に関連づけて，演奏者自身の判断で数値化して与えておく．さらに楽曲 (チューン) としてコード進行に従った伴奏データのファイルを用意し，その伴奏の上で，適当なアドリブ・フレーズを順次送り出すことによってアドリブを自動生成させる．

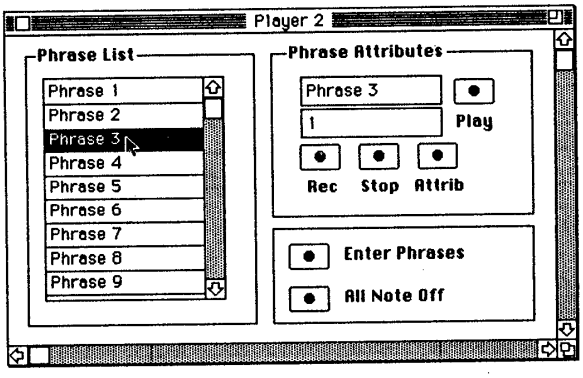

図 1: Player Window

3　BoP 2 の実装

BoP 2 の開発は，Apple 社の Macintosh 上において，Symantec 社の C++コンパイラを用い，GUI 環境のプログラミングの利便性を考慮して同社の提供するアプリケーション・フレームワークである THINK Class Library 2.0(以下 TCL，または TCL 2.0) を基礎に置いて行った．TCL 2.0 は，Symantec C++ for Macintosh の Version 7 以上の環境で動作し，C++のオブジェクト指向プログラミングを実現している．さらに，Visual Architect と名付けられた GUI 構築用の，プログラムの自動生成トゥールが用意されているため，GUI 構築の省力化が行える．アドリブ・フレーズのテーブルであるプレーヤーを表すウィンドウを図 1に示す．

Macintosh 上で MIDI プロトコルを扱うために，Apple 社が提供している MIDI Management Tools を用いる．MIDI Manager をコントロールするために，Symantec C++ 6.0 と TCL 1.1.3 と整合性を持つ CMIDI class library を採用し，Symantec C++ 7.0 と TCL 2.0 上で動作させることができ

Implementation of Automatic Ad Lib Generating Program in Light Music
Yoshiro KISHIDA, Tsunetoshi HAYASHI
Department of Computer Sience, Faculty of Science and Engineering, Ritsumeikan University

るように修正を加えた.

TCL においては, プログラムのクラス・ライブラリの木構造の構成は大きく分けて, プログラムの中核をなすひとつのアプリケーション部と, データ構造の処理を行うドキュメント部, そして GUI を含む視覚的な部分の処理を行うヴュー部, さらにその他の部分に分けられる. BoP 2 においては, 他に CMIDI クラス群が加えられる. CMIDI クラス群には, MIDI Manager の Patch Bay 上にアプリケーションをひとつのクライアントとして登録させるためのクラスと, 時間, 入力, 出力の各ポートを持たせるためのクラス群が含まれる.

MIDI 信号の処理の重要な部分は, アプリケーション部によって行う. アプリケーション・クラスに CMIDI の入・出力ポート・クラスのインスタンス変数を持たせることによって, MIDI 信号の入出力を行う. MIDI パケットの時間管理は, MIDI Manager のパッチベイに対して不可視な出力ポートに送られた MIDI パケットを不可視な入力ポートに送信し, そこでタイムスタンプを評価させることによって行う. さらにそのパケットは通常の出力ポートに送られて外部の MIDI 機器に送信される.

ドキュメント部では, MIDI データの処理に関するメソッド類と GUI の統括を行うメソッド類の実装を行う. ドキュメント部は, さらに 2 種類のドキュメントに分かれ, それぞれ演奏家と楽曲のデータを受け持たせる.

演奏家のドキュメントであるプレーヤー・ドキュメントは, アドリブ・フレーズの打ち込み, あるいは MIDI 信号によるリアルタイム録音を行い, また, ファイルのオープン, 保存などの処理を行う. また. 楽曲のドキュメントであるチューン・ドキュメントは, コード進行についての同様の処理を行う.

4 アドリブの生成

アドリブ・フレーズの選択は, BoP 1 と同様, 各フレーズにターゲット領域を持たせておいて, 発生させた乱数がどのターゲット領域に入るかで行う [3]. ターゲット領域の大きさは, 相性度 (P) によって定められる. 相性度 (P) は, アドリブ・フレーズの入力時にその属性として, 主としてコードとの相性の程度を演奏家自身が指定しておく. その値は 0 から 100 までの数値として入力されるが, フレーズの開始が小節中のどの拍になるか, また, down beat もしくは up beat のどちらから開始するのかなどによっても, 分けて入力されるようにしておく.

相性度 (P) は, 楽曲中の位置やコード等の楽曲の進行状況による重みづけを行ってから出現確率の算出に用いられる. 例えば楽曲の開始時には, 開始時を好むフレーズの相性度 (P) を大きくする. また, フレーズのつながりが, 近い音程が良いのか遠い音程がよいのかなどの重みづけを行う. また, 曲中の盛り上がりの情報を参照してフレーズの相性度 (P) の重みに変化を付ける.

相性度 (P) の値は, 2 乗して出現確率の算出に用いることにより, 演奏家の実感に近い自然な確率が得られる [3].

5 終わりに

今後, より優れた, さらにユーザ・フレンドリーな GUI を構築して, アドリブ・フレーズの入力や, 楽曲の入力に際して, ミュージシャンの繊細な要求に答えられるようなプログラムの開発を進める必要がある. また別の方法として市販のシーケンシャル・ソフトの豊富な機能の利用を考えると, 市販のソフトとのアプリケーション間の MIDI プロトコルによるリアルタイムな通信の実装を行うことを考える必要もある.

参考文献

[1] Max V. Mathews and John R. Pierce: "Current Direction in Computer Music Research", The MIT Press, 1989.

[2] 特集: 音楽情報処理, 情報処理, Vol. 35, No. 9, 1994.

[3] 岸田良朗: "知識情報に基づく, 軽音楽におけるアドリブ演奏の自動生成", 第 50 回 (平成 7 年前期) 情報処理学会全国大会講演論文集 (分冊 1), 1995.

[4] Paul D. Ferguson: "MIDI Manager Class Library for Symantec C++ Version 2.2 Programmer's Guide", 1994.