

自動表情付システム MorPH の表情付記述言語

中口 孝雄, 樋口 直史, 真栄城 哲也
ATR 人間情報通信研究所 第6研究室

要旨

ポップス系の自動表情付けシステム MorPH の表情付けを記述するルール形式の言語を開発した。ルールは条件部と実行部から成り、条件部には任意の論理積のみを用いた記述が可能だが、実行部には用意されたものしか利用できない。条件部に他のパートを参照することで、曲全体として一体感のある表情付けが可能である。ドラム、ベース、ギターのソロおよびバックিং、そしてサックス用のルール集合を用意した。楽器毎に複数のルール集合を用意することで、演奏の個性が強くなる部分にも適用可能である。

Musical Performance Description Language for MorPH

Takao Nakaguchi, Naofumi Higuchi and Tetsuya Maeshiro
Dept.6, ATR Human Information Processing Res. Labs.

Abstract

A language to describe the performance humanization process of MorPH is presented. The syntax of the language is a rule, consisting of a condition and an execution part. For the condition part, multiple conditions are associated only by logical AND. Only the provided execution commands or functions are allowed in the execution part. Reference to other instruments' parts in rules enables unified performance data. Rule sets for drums, bass, guitar solo and backing, and saxophone are provided.

1 はじめに

楽曲の表情付システムとして、種々発表されている [1]。本研究で対象としているのは、MorPH (Musical Performance Humanizer) [2, 3] は、ポップス系の楽曲の SMF データの自動表情付システムであり、インターネットで公開中である (<http://morph.hip.atr.co.jp>)。ブラウザからアクセスする “MorPH light 1mg” と、Windows 95/98 専用のクライアント “MorPH Rhapsody” の 2 種類が存在し、どちらもユーザーが指定した好みのドラムパターンを基に、曲に合ったドラムパートを自動生成し、ドラム以外の楽器の自動表情付けを行う。“MorPH Rhapsody” では、ドラムパートのゴースト音や、フィルインの位置や細かさ等が指定可能だが、ドラム以外の楽器に関しては、表情付け処理をするかしないかの選択しかできない。また、これらの楽器の表情付けは、あらかじめ定められた方法で行われ、「演奏家 A 風」、「演奏家 B 風」のように異なった表情付けができない。さらに、ソロパートの様な演奏の個性が顕著に出る場合には、対応できない。

このような多様な演奏方法と好みに対応するため、楽器毎の表情付けをルールで記述し、そのルールを選択することで表情付けを行うように MorPH を拡張した (図 1)。ルールの編集を一般ユーザーに公開することで、より使い勝手の良いシステムとなることが期待される。なお、事例ベースではなく、ルールベースとした理由は以下の通りである。(1) 事例ベースの場合、MorPH が対象とする MIDI データでは、実演奏のデータが少なく、事例の収集が困難である。(2) 事例ベースの場合、MorPH が目的とする曲全体としてまとまりのある表情付には、複数パートを含む事例を入力する必要があるが、パート間の関連性の特徴抽出および類似度の定義が困難である。(3) 学習機構の組み込みが容易である。

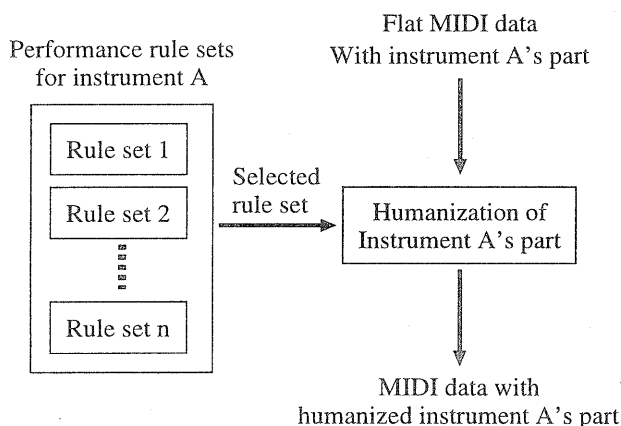


図 1: 1つの楽器の処理。左の rule set 1 から rule set n のうちの 1つのセットをユーザーが選択し、そのルールを用いて表情付けの処理を行う。

第 2 節では、表情付け言語について述べ、第 3 節で、考察および結論を述べる。

2 表情付け記述言語

MorPH の表情付け記述言語は、単純なルール形式であり、言語仕様は可能な限り単純にしている。

2.1 ルールの仕様

ルールの記述は、**IF** で始まる条件部と、**DO** で始まる実行部から成る。各楽器、もしくは MIDI のプログラム・ナンバー毎に、複数のルールの集合を用意する。図 2 は、ベースのビブラートのルールの記述例である。条件部の記述には、論理積 (AND) のみが許されている。従って、論理和 (OR) を使う必要がある場合には、条件を展開し、個別のルールとして記述する必要がある。同様に、条件の記述の階層化もできない。実行部には、「ノートにビブラートをかける」、「拍のドラムパターンをフィルインにする」などの処理内容を記述

する。ルールの記事を、論理和を含まない独立形態にしたのは、学習機構を組み込みやすくするためである。現時点では、ルールの条件部のみ編集可能にしている。MorPH には、基本的な処理は組み込まれているため、それらの選択と条件の組合せで好みや演奏方法の多様性に対応できると考える。

条件部分に記述可能な対象は、MorPH が扱う曲の構造単位であり、ブロック、フレーズ、小節、拍、ノートの羅列、である。これらの構造単位中の、前 / 後ろから i 番目のノートといった指定も可能である。

```
IF Track.InstrumentType == 'Bass' and Track.InstrumentName != 'SlapBass1'  
and Beat.PosInMeasure == 1 and Note.ContinueToNextBeat != true and  
OffTimeInMeasure <= Range(1440, 1920) and Note.DeltaPitchToNext >= 7  
DO BassProc.slide
```

図 2: ベースのビブラートのルールの記述

表 1 は、表情付けのために用意されている処理の一覧である。これらは、MorPH の内部処理に用いられていた処理である。

2.2 表情付け処理

基本的に、ルールの集合が与えられると、まず全てのルールの条件部分を処理対象の MIDI データに対してマッチングを行い、マッチしたルールを用いて処理を行う。このマッチングと処理の繰り返しによって表情付けは処理されるが、マッチするルールが無くなるまで繰り返される。現在、ループ回数に上限を設けて、無限ループを回避するようにしている。

条件部分には、対象とする楽器のパートの情報だけではなく、他のパートの情報を組み込める。例えば、MorPH のドラムパートのルールは、通常のドラムパターンの生成にはベースのパートを、フィルインの生成には他の全てのパートを参照する。他のパートを条件に記述することで、曲全体としてまとまった表情付けのルールの記述が可能である。

2.3 ルールの競合

ルールの競合は、ルールの条件部分ではなく、処理部分によって生じる。基本的には、より狭い範囲を処理するルールを優先的に実行する。これは、処理範囲が狭い程、特徴的なルールである可能性が高いという仮定に基づく。また、処理対象が同じ場合、条件の多いルールを選択し、それも同じならば、ランダムにルールを選択する。

MorPH のように MIDI データを扱う場合、2 種類のルールの競合がある。(1) ピッチバンドやエクスプレッションの値を生成 / 変更する処理の場合、それらの値が生成された範囲の音符 (ノート) は全て影響を受ける。この場合、同範囲に対して他の処理は行えない。(2) ノートのベロシティの変更やノートの追加等、同範囲の他のノートと干渉しない場合、

表 1: 各楽器のルールの実行部分に使用可能な処理

楽器	処理
ドラム	通常拍のドラムパターン生成 フィルイン拍のドラムパターン生成 フィルインの最後の拍のドラムパターン生成
ベース	発音時間の設定 ハンマリング ダウンスライド ビブラート ゴースト音の追加
ギター	チョーキング ダウンスライド ビブラート ハンマリング ゴースト音の追加 コードのストローク アルペジオ処理
サクソフォン	通常のエクスプレッション・カーブ 吹き上げ 連続音

である。特に、他と干渉する (1) の競合は注意を要し、そのような処理が行われた場合、他のルールの処理を禁止する。ルール競合はできるだけ回避できることが望ましいが、全て回避できるものではなく、最終的にはユーザーのルール記述 (プログラミング) 能力に依存する。

3 考察および結論

表 2は、MorPH の初期バージョンで、サーバーのプログラムに組み込まれていたルールを明示的に記述することで抽出された楽器毎のルール数である。これらのルールは、大部分は基本的な表情付に関わるものであるが、著者らの好みの演奏方法を記述するルール集合であると言える。これらのルール集合は、動作が保証されているため、ルールの競合は生じない。ルール競合の解消に関しては、さらなる検討が必要であり、ルール編集機構を組み込んだ MorPH の一般公開後、ユーザーのルール集合によるテストを行う予定である。

また、ルールの不足により、表情付け処理されない部分が出現するが、そのような部分

表 2: 楽器毎のルール数

楽器	ルール数
ベース	51
ドラム	5
ミュートギター	64
それ以外のギター	6
サクソフォン	13
計	139

は未処理のままにしてある。別の選択肢として、未処理部分はシステム内部に用意してあるルールで処理することも考えられるが、表情付けの好みの問題と、ルールの条件部ではなく、ルールの実行部によってルールを選択する機構が必要になるため、検討を要する。ルールの記述は、完全に人間が行う必要があり、ルールの自動抽出の検討が必要であるが、複数パートの楽曲での特徴抽出が課題であるが、実演奏の MIDI データの収集から始める必要がある。

現在、ルールの実行部には、MorPH で用意している表情付処理しか使用できず、新たな表情付処理の追加をユーザーに開放することは、予定していない。ユーザーによる処理の追加機構より、例えばノート単位の表情付のような基本的な表情付処理を多く提供する方が良いと考えている。

ルール記述機能として、GUI を用いたものを予定している。表情付言語の仕様も、ユーザーの反応によって、条件部に論理積以外の論理演算子の追加や、実行部に使用可能な表情付処理の増加も検討する。

また、クラシファイアシステム [4] を用いた表情付けの学習機構を組み込む予定である。

謝辞

下原室長のサポートを感謝する。

参考文献

- [1] 平賀 瑠美 (1998) “演奏の表情付け”, コンピュータと音楽の世界, 共立出版.
- [2] 真栄城 哲也, 中口 孝雄, 樋口 直史 (1999) “自動表情付システム MorPH”, 情報処理学会研究会報告, 99-MUS-11, 99:67-72.
- [3] 中口 孝雄, 樋口 直史, 真栄城 哲也 (1999) “自動表情付システム MorPH の実装”, 情報処理学会研究会報告, 99-MUS-11, 99:61-66.

- [4] D. E. Goldberg (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.