

# TMULE : 楽典を考慮した音楽記述言語

naegawa(@onuxy)

ダウンロード先 : <http://nlab.flnet.org/tmule/index.html>

## 1 はじめに

近年、ロボット技術研究会でも音楽に興味を持つ人達が増えてきたように感じます。ロボット技術研究会は「ものづくり」のサークルであるので音楽も作ってみたいと思う人も多くいるでしょう。そこで、もっといろんな人に楽曲制作を楽しんでもらうために音楽記述言語 TMULE (Theory-based MUsic LanguagE) を開発しました。TMULE は楽典の知識を持たずとも言語にそれらの知識を言語に内包することにより簡単に理論に基づいた作曲を行うことができるようにした言語です。また、TMULE は作曲の初期段階を意識して製作されており、TMULE で書かれた楽曲は標準 MIDI ファイルとして出力可能です。

## 2 TMULE の基本概念

TMULE では以下の3つのステップを持って音を構成します。

- 音階 (scale) 記述
- 音階上での和音 (chord) 記述
- 和音上での旋律 (melody) 記述

まず、音階 (scale) 記述では音の並びを指定します。ここでの音の指定は MIDI における MIDI ノート番号を用いており、60 が中央のドを表し 61 がド #、62 がレ、63 がレ #、というようにピアノの鍵盤で言うと、白鍵・黒鍵関係なく順番に番号がついています。また音階の始まりの音を主音 (tonic) と言います。ハ長調の音階は次のように記述します。(改行は無視されます)

```
scale MajorScale(tonic=60){  
    0(100),2(100),4(100),5(100),  
    7(100),9(100),11(100)  
}
```

MajorScale の部分は音階名で自由に名前をつけることが可能です。tonic=60 の部分は主音を指定しています中括弧の中の数字は主音からの相対的な音程を表し、0 ならば 0+60 で 60 つまり中央のドを表します。丸括弧内の数字はデフォルトの音の強さ (0~127) を記述します。

次に、和音 (chord) 記述ではスケールの内 1 フレーズ内でどの音を用いるかを記述します。和音で音階に基づいて音階の何番目の音かということ指定します。以下に 7th コードの例を示します。7th コードには主にメジャー 7th、ドミナント 7th、マイナー 7th がありますが、この指定方法だとスケールに合わせて自動的に決定されます。

```
chord Seventh(scaleBase){  
    0,2,4,6  
}
```

Seventh の部分は和音名で自由に名前をつけることができます。scaleBase の部分は音階に基づいて和音を決定するということを指定しています。(別の指定方法については後述) 中括弧の中の数字は音階上で何番目に位置するかを指定しています。(楽典の「度」とは異なり 0 から始まることに注意) 例えば、先に定義した MajorScale で Seventh を鳴らすとそれぞれ 60,64,67,71 となり、ド・ミ・ソ・シとなります。

最後に旋律を記述します。旋律は和音の何番目の音かを指定して決定します。以下に旋律の例を示します。

```
melody Mel1(chordBase){
  {0 ,r ,1 ,_ },
  {2 ,3 ,_ ,_ }
}
```

Mel1 の部分は旋律名で自由に名前をつけることができます。chordBase の部分は和音に基づいて旋律を決定するというを示しています。(別の指定方法については後述) 2 重中括弧の部分が具体的に旋律を記述した部分です。数字は和音の何番目の音かを表し、r は休符を表します。また、\_(アンダースコア) は前の音を伸ばすという意味を表します。つまり、MajorScale、Seventh のときの 0 ,r ,1 ,\_ はドの音を鳴らしたあとに休符が入り、ミを最後まで伸ばすという意味になります。また中括弧を 2 つ並べて書くと、それらは同時に演奏されます。したがって MajorScale、Seventh のときの Mel1 を別の見方をすると、「ドとソの音を鳴らしたあとにシの音を鳴らし、シを鳴らし続けながらミを鳴らして最後まで伸ばす」という意味になります。

以上の方法で音を構成した後に実際にこれらを指定して曲を作成します。曲の作成にはトラックの記述により行います。トラックでは使用する音階・和音・旋律を指定して、それらを並べる事により記述します。ただし和音と旋律は角括弧でペアにして指定します。以下にトラックの例を示します。

```
track Tr1(){
  MajorScale,
  [Seventh,Mel1],
  [Seventh,Mel1]
}
```

Tr1 の部分は旋律名で自由に名前をつけることができます。この例では音階に MajorScale を指定してその後 Seventh の和音で Mel1 の旋律を 2 回繰り返すという意味になります。別の音階 MinorScale を定義すれば、以下のように途中で音階を変更する

ことができます。

```
track Tr1(){
  MajorScale,
  [Seventh,Mel1],
  MinorScale,
  [Seventh,Mel1],
}
```

ここまでの例では基本的な指定方法しか説明しませんでした。和音記述においての音階の指定を scaleBase ではなく scale12Base とすることで 12 音すべての音を持つような仮想的な音階で指定できます。また、旋律記述において chordBase ではなく dcaleBase を指定することで音階上の音を自由に指定することができます。

### 3 TMULE とプログラミング言語

TMULE は (C 言語などの) プログラミング言語を強く意識して作られています。音階・和音・旋律の各記述は関数とみなすことができ、音階の主音の指定などでは引数とデフォルト引数という概念を用いて記述する量を減らしています。また、音階・和音・旋律は型に対応させることができます。このような対応により今後更に柔軟な記述が可能になると考えています。

音楽記述言語にはプログラミング言語と同様の利点が他にもある。プログラミング言語ではコンパイル時にエラーをチェックすることにより実行時の致命的なエラーを防いでいる。同じく、音楽記述言語においてもコンパイル時にエラーをチェックすることにより作った曲が理論に従っているかどうかをチェックすることができる。ただし、音楽の場合は理論に例外が多く、また意図的に理論から外れたことをすることも多いためコンパイル時にユーザに警告を発する程度にするべきである。

### 4 他の音楽記述言語との比較

音楽記述言語として古くから使われている言語に MML(Music Macro Language) がある。MML ではテキストを用いて楽譜やシンセサイザー設定を

記述する。しかし、可読性がそれほどよくないという問題がある。日本語で音楽を記述できる環境としてテキスト音楽「サクラ」があり、ストン音階という独自の表記法により音楽を表現している。ただし、内容的には MML とほぼ同等である。

MML を発展させた PMML ではアクセントやクレッシェンドといった音楽的表現やスレッドという複数パートを記述するための方法を導入してこれらの問題を解決している。さらに、PMML ではレジスタと呼ばれる設定値や変数・関数・演算子といった概念もありプログラミング言語に近いものとなっている。その他にも Formula、Common Music といった汎用プログラミング言語の拡張として作られた音楽記述言語がある。これらでも変数・関数を駆使することにより TMULE と同様の記述を行うことができるが、言語仕様としているわけではない。したがって、理論に合っているかのチェックなどを行うことは簡単ではない。

プログラミング言語との関連で見ると、MML などの音やシンセサイザーの設定と一対一で対応している言語は機械語と一対一で対応しているという点でアセンブリ言語と似ている。また、Formula ではスタック構造を利用して音を鳴らすという記述により和音を鳴らす工夫をしている。これは JavaVM などで見られるスタックマシンと似ている。

また、データを記述するという側面から音楽記述言語を見た場合には他にも XML ベースの記述言語が多く存在している。これらは GUI から機械的に出力する場合やストレージに保存する場合、Web や印刷で表示するために用いられている。TMULE は楽曲を作るということに焦点を当てているためこれらの言語とは異なっている。

## 5 他の音楽開発環境

作曲を支援する方法としては音楽記述言語の他にも様々なものがある。ここではよく使われる方法の一部を紹介し、音楽記述言語との関係を示したい。

### 5.1 MIDI シーケンサ・DAW

近年の MIDI シーケンサは DAW と一体となっているものが多いため、ここでは同時に紹介する。

これら 2 つの機能を持つものとしては Cubase や Logic、SOL (XGworks)、Cakewalk などがある。MIDI シーケンサは音楽記述言語とは異なり MIDI を直接編集できるといった点や GUI が整備されているという点で非常に優れている。TMULE では、大雑把に作曲した曲を MIDI で出力しそれを MIDI シーケンサで仕上げ、さらに DAW で加工・レコーディングといった作業をすることを想定・推奨している。

### 5.2 ループシーケンサ

近年ループシーケンサと呼ばれるループ・フレーズ素材を組み合わせて曲を作るという方式の作曲ソフトが非常に増えている。例としては ACID、Music Maker MX、Cakewalk PLASMA、GarageBand などが挙げられる。思い通りの作品を作るには非常に多くの素材が必要という欠点はあるものの、これらは素材を組み合わせて非常に簡単に曲を作ることができる。また、自動作曲についてもかなり実用的な方法が存在する。音楽記述言語を用いてフレーズを作成し、これをループシーケンサの素材として用いることは十分可能であると考えられる。

## 6 まとめ・今後の発展

音楽は古くから研究されているため多くの理論が存在する。今回、開発した TMULE はその内のごく基本的な部分しか考慮できていない。また、TMULE がターゲットとしている作曲に関しても多くの理論が存在している。さらに、自動作曲の分野では今も多くの理論が研究されている。これらの内、音楽記述言語に取り入れるにあたって有用な理論というのは明確ではない。しかし、理論を言語として取り入れることにより、その理論をより活かすことができるだろう。今後そのような理論を見つけ次第、実装していきたいと考えている。