

## 電子音響音楽の構造的制作方法－「Four tears」を例として

常盤 拓司

慶應義塾大学大学院政策・メディア研究科

tokiwa@sfc.keio.ac.jp

### 概要:

筆者の制作した電子音響作品「Four tears」を例に、作曲のためのツールを利用して作曲する方法について検討を行った。作品制作では、Csound と、楽譜ファイルの生成に作曲ツールの一種 CMask を用いた。CMask で楽譜ファイルを生成する際、楽曲を複数の音から作られたブロックの組み合わせとして構成し、各ブロックを CMask で生成した。その結果、楽曲の構成を検討するための基準として個々の音や音符を見るのではなく、音のブロックを、楽曲を構成する単位の大さきの基準として用いることがコンピュータを用いた音楽制作に有効であろうという結論に至った。

キーワード: コンピュータ音楽、アルゴリズム作曲、音楽構造

## A structural composition method of electro acoustic music – a study of “Four tears”

Takuji Tokiwa

Graduate School of Media and Governance Keio University

tokiwa@sfc.keio.ac.jp

### Abstract:

In this paper, about electro acoustic music piece “Four tears” which is composed by author, the method for composition which is supported by composition tools was considered. The piece was made by Csound and CMask which is a composition tool. Pieces were regarded as suite of blocks which is made by sounds when CMask generates score files. It is an effective method that sound blocks would be regarded a standard size of element of music piece, for composing with computer.

Keywords: Computer music, Algorithmic composition, Music structure

### 1. はじめに

本稿は、筆者の制作した電子音響音楽作品「Four tears」での、作曲ツール CMask を用いた制作の過程において用いた手法について検討を行ったものである。本稿でこのような検討を行うのは、筆者が常にコンピュータを用いて音楽作品を制作する際に感じる、音楽構造の曖昧さがもたらす制作作業の混乱が、「Four tears」の制作の際に発生しなかったということが動機となっている。

ここで、音楽構造のあいまいさ、とは、楽譜による音楽のように構造を構成する単位が明確な音楽に対し、電子音響音楽のようなタイプの音楽ではそうした構造がしばしば明確に存在しない、ということである。

楽譜による音楽の場合、最小構成単位は、音符の一つ一つである。それらが集合してモチーフや、メロディーなどといわれる楽曲を構成するための機能を持つ構成単位が形成される。そのような単位でとらえ

られる構造の複合体として楽曲をとらえることができる。

しかし、電子音響音楽のような音符によらない音楽形式の場合、音楽の最小構成単位は、例えば、ある持続時間(duration)で鳴り続ける、楽譜やグラフ上に構想された音、あるいは録音された音などのこともある。録音された音が具体音のような“物語”のある音の場合、その“物語”はさらに細かな単位によって構成されているということがありえる。

また、“テクノ・ミュージック”のような、一部のポピュラー音楽の場合、構成単位のあいまいさはより顕著である。こうした音楽はしばしば、シンセサイザーやサンプリング・マシンなどを利用し、パターンを繰り返して重ね合わせて構成される。

シンセサイザーのキーを一つ押すだけで、アルペジオが自動的に発生するような仕組みや、リズムパターンがひとつの音としてサンプリング・マシンの一つ一つのキーに割り当てられ、それらを、タイミングにあわせて呼び出すというのはごく一般的な制作の手法である。アルペジオやサンプリング・マシンに仕込まれたリズムなどは、明確に聞き分けることのできる音程とリズムの異なる音によって構成されていることが多い。したがって、その最小構成単位は音符であると考えることが可能である。しかし、それを発現させる行為は、キーを一個押した、あるいは、MIDI データを一個送るということではない。それを仮に楽譜に記述するならば、機械的に作り出されるアルペジオの音の連なりも、サンプリング・マシン上のリズムパターンも、ひとつの音符、あるいは記号ということになる。これはシンセサイザーや、サンプリング・マシンなどに限ったことではなく、実験音楽などのチャンス・オペレーションや、パーカッションの演奏指示などにも当てはまることである。そして、さらに別の見方をすれば、「アルペジオというイベント(あるいはプロセス)を発現させよ」、あるいは「リズムパターンを1回ならせ」などというような命令・トリガーということができる。そしてこの命令を最小構成単位と言うことは、不可能なことではない。

これらから、最小構成単位は、必ずしも、音符や音ということで無ければならないということではなく、「機能の発動」と、捕らえることも可能な場合が存在しているということができる。

これはつまり、音楽の構成単位は 2 種類あるということの意味している。一つは分析と記述のための単位としての音である。もう一つは、動作を指示するための単位である。

そしてさらにコンピュータ音楽にまで視点を広げた場合、アルゴリズム作曲のためのツールなどによって作り出される音符や音の連なりは、先に述べたアルペジオのようなパターンが繰り返されるとは限らない。また、Granular synthesis のような、音の連なりが、一つの音となる場合もある。

本稿で取り上げる「Four tears」の制作にはコンピュータとソフトウェアを用いている。このことに注目するならば、コンピュータを用いて音楽を制作する際に筆者が扱う単位の一つは、ソフトウェアに対する命令だということができる。これを「Four tears」の制作に限って言うならば、「音を生成せよ」という命令になる。しかし、この命令は、さらに別のソフトウェアとそのソフトウェアに与えた命令からできている。

このような場合、最小構成単位、あるいは音楽の構造を捕らえる為の単位は一体どこにあるのだろうか、あるいは、そもそも最小構成単位という概念自体が成立するのだろうかという疑問と、楽曲の制作作業をどこから行うべきなのか、あるいはどこまでを制作作業と呼ぶべきなのかという疑問を感じさせられる。

筆者は、これまでコンピュータを用いて電子音響音楽の作品を制作する際、このような最小構成単位のあいまいさによってもたらされる、作業の混乱とでもいうような疑問に直面してきた。ところが、「Four tears」の制作においては、このような混乱がまったくおきなかった。

筆者は、混乱がおきなかった要因は、この作品を制作するために採用したソフトウェア環境(Csound と CMask)がもたらした、音をブロック単位で扱うという方法が、最小構成単位のあいまいさを結果的に遠ざけたためではないかという内観を持っている。

本稿は、筆者の作品「Four tears」の制作過程で用いた、このブロック単位での楽曲制作の方法について検討を行い、作曲ツールの利用する場合に、最小ではないが有効な楽曲の構成単位について述べる。

## 2. Four tears について

「Four tears」は 4 つの楽章からなる電子音響音楽作品で、これまでに、First movement (第 1 楽章)と Third movement (第 3 楽章)が上演されている。(表 1)

「Four tears」は、ユニットジェネレータ言語の一種である Csound と Csound での作品制作を支援するた

めに開発された楽譜ファイル生成用のソフトウェアの一種、CMask によって制作された、“テープ作品”である。

Four tears の制作に、CMask を用いたのは、楽曲の制作に確率的要素を採り入れたいというごく単純な理由からであった。

Csound は、「作曲家が楽音合成オーケストラ(orchestra)とそのオーケストラを参照する楽譜(score)を作る」という概念に基づいて設計されている。オーケストラの中の楽器(instrument)は、テキスト形式でパッチされる。すなわち、テキストによりユニットジェネレータ(unit generator)間の相互接続を指定し、ユニットジェネレータの入力パラメータを決める、各ユニットジェネレータは、オーディオ信号や制御信号(エンベロープ)を発生したり、それらの信号を偏重するソフトウェアモジュールである。楽音合成プログラムやコンパイラは、オーケストラファイルと楽譜ファイルを読み、機械レベルのパッチの表現を構成し、そこに楽譜ファイルのデータを与える。最終的なオーディオ信号はサウンドファイル(sound file)に書き込まれる。[1][2]

表 1 「Four tears」の発表歴

発表タイトル	コンサート名	開催地	日時	特記事項
Four tears -first movement	インターカレッジコンピュータ音楽コンサート 2001	慶應義塾書何藤沢キャンパス	2001 年 12 月	
	11th Annual Florida Electro Acoustic Music Festival	University of Florida (アメリカ)	2002 年 3 月	入選
	MAXIS A Festival for Experimental Music and Sound	Sheffield Hallam University (イギリス)	2002 年 4 月	入選
Four tears -third movement (for friend)	Concert “Ear-popping Sounds!”	Theater im Depot (ドイツ)	2002 年 6 月	F. Niehusmann 氏のキュレーションによる

CMask は、正式には、“CMask a stochastic score generator for Csound”という名称で、Andre Bartetzki 氏によって開発された、楽譜ファイルの制作を支援するためのソフトウェアの一つである。

CMask は、スクリプトによって定義される生成プロセスにより発生する確率的な数値をもとに、Csound の楽譜ファイルに記述されるパラメータを作り出す。

生成プロセスは、原理的には、なんらかの方法で数値列を発生させ、tendency mask, quantizer, accumulator という直列に接続された 3 種類のフィルターを適用していく。数値列の発生は、確率分布の形状を選択することが可能な乱数発生器、もしくは VCO 的に数値を循環的に生成する発振器によって行われる。(図 1)

CMask のスクリプトは、フィルターに入力する数値をどの方法で生成するかや、どのフィルターを用いるか、フィルターに与えるパラメータの時間軸上での推移を定義することができる。

### 3. Four tears の制作方法

本稿で取り上げる「Four tears」の二つの楽章の制作においては、複数の音からなる“ブロック”を、楽曲全体を構成するための単位とするという方法を用いた。

制作当初、この方法は単に、Csound と CMask を用いて制作する際に出てきた制作作業上の問題を解決するためのものであった。作業上の問題とは、楽曲全体を一つの CMask のスクリプトで制作すると、スクリプトが読み難くなり、楽曲のイメージがつかみにくくなるということであった。これを解決する方法として、楽曲全体を複数の CMask のスクリプトのブロックが組み合わされたものとして捕らえ、楽曲全体を、これら

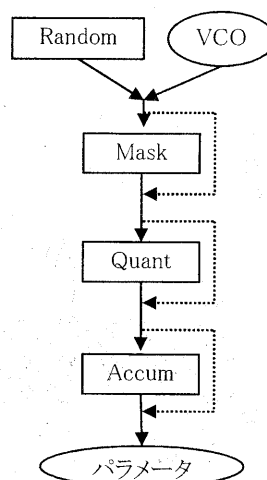


図 1 CMask の流れ[3]

の複数の CMask のスクリプトの組み合わせとして考えるという方法を用いた。

その結果、「Four tears」では、始めに全体の構造をブロックの配置として考えてから、CMask のスクリプトによって、各ブロックを生成するという方法が、楽譜ファイルの制作(楽曲の構成の制作)の基本戦略となった。

#### 4. First movement 制作の流れ

実際の、First movement の制作では、先の基本戦略に加えて、楽器(オーケストラファイル内に記述される)の制作が必要となる。そのため、この楽章の制作では、初めに音色(すなわち楽器)の制作から作業を行った。

この楽章では、(a)低速で再生した川の流れる音の断片と、(b)(a)に楕円形フィルターをかけて得られた音の2種類の音色を用意した。(a)低速で再生した川の流れる音の断片は、30秒ほどの川の流れる音から、音を部分的に切り出し、インターポレーション無しで、通常速度の1/2から1/4程度の低速で再生するという方法で生成される。そして、(b)(a)に楕円形フィルターをかけて得られた音は、(a)と同様の方法で切り出した音に対して、楕円形フィルターをかけて作り出している。

この音色の制作の後に、CMask のスクリプトによって生成するブロックを単位として、楽曲の構成を方眼紙上に手作業で行った。(図2)

そして、このブロック単位での楽曲の構成に基づいて、CMask のスクリプトとして各ブロックの制作を行った。CMask による各ブロックの生成では、そのブロック内での発音開始時刻や、音の長さ、サンプルファイルの再生開始位置、楕円形フィルターのフィードバックの遅延時間(聴こえ上の音の高さが決定される)、定位などの、オーケストラファイル内に記述された楽器が音を生成するために必要なパラメータが、確率的に生成されている。

各ブロックを構成する音は、20秒から70秒程度で、ゆっくり減衰していく。また、楕円形フィルターによって作られたブロックを構成する各音は等間隔の倍音列をもつため、倍音列が重なった場合、強めあったり、打ち消しあったりということが発生する。その結果、アタック以外の要素で個々を明確に分けて聞き取ることが難しくなり、各ブロックは、明確な音の構造を感じ取りにくい、音のかたまりとなっている。

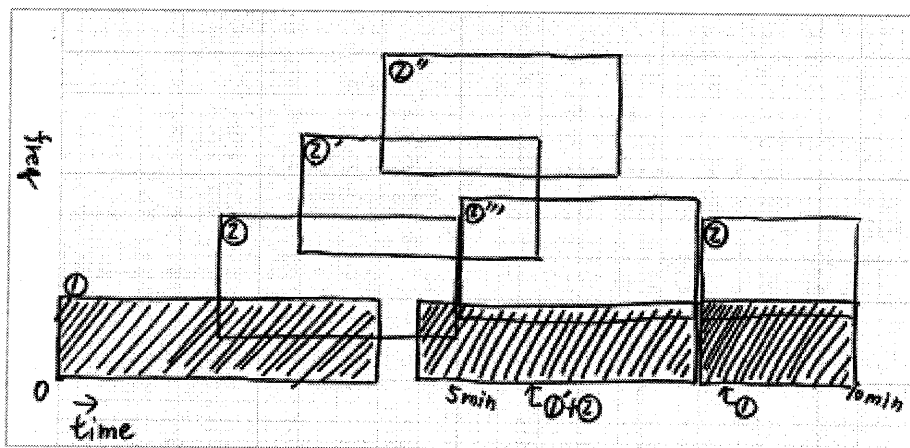


図2 First movement の全体構成(実際のスケッチ)  
(斜線のブロックが音色①、枠のみが音色②)

#### 5. Third movement 制作の流れ

Third movement の場合、First movement と同様に、(1)音色の制作→(2)楽曲の構成→(3)CMask による各ブロックの生成、という3つのプロセスで制作を行った。しかし、First movement の場合と異なり、ブロックごとに独立したサウンドファイルをつくり、最後にミックスダウン用のオーケストラファイルと、楽譜ファイルを

用いて、ミックスダウンを行った。

このような方法を用いているのは、各ブロックを生成するための楽譜ファイルが、短いもので 3000 行程度、長いもので 7000 行程度になり、サウンドファイルの生成に時間がかかるためであった。

このような長さになった理由は、Grain を 1 つだけ生成する“楽器”を、楽譜ファイルで 1 秒当たり 20～40 回程度呼び出して、Granular synthesis を実現しているためである。これは、Csound にすでに実装されている Granular synthesis の関数では、筆者の音楽的な要求を満たすことができなかったためである。

この楽章では、教会の鐘を一回だけたたいた音と鐘が何度もたたいた音を用いている。前者は Granular synthesis を行う際に使用し、後者は、楽曲の最後に使用している。

この鐘の音のための楽器の制作の後、First movement と同様に、CMask で生成するブロック単位で楽章の全体を構成しているが、First movement とは異なり、方眼紙を用いず、ミックスダウン用の楽譜ファイルとして、各ブロックの時間的な位置の構成を行った。これは、この段階ですでに、各ブロックをそれぞれ独立したサウンドファイルとして生成することが分かっていたため、楽曲の構成を初めから楽譜ファイルとして記述したほうが効率的であるという判断を下したためである。

そして、この全体の構成に基づいて、各ブロックを、CMask のスクリプトを作成し、楽譜ファイルによる Granular synthesis を行っている。

## 6. ブロック単位での楽曲構成についての考察

本稿でとりあげた、楽曲「Four tears」の First movement および、Third movement の制作プロセスでは、CMask による楽譜ファイルの生成を効率よく行うという観点から、CMask のスクリプトをブロック単位で分けて記述するという方法を用いた。これが結果的に、筆者の感じてきた作品制作の際の混乱の原因である最小構成単位の曖昧さを遠ざけたと考えている。

それはつまり、CMask のスクリプトのブロック、言い換えれば、ある音の連なりを生成する仕組み、あるいはそれによって実際に生成された音の連なりが、楽曲を構成するための単位の基準として機能したからである。これはブロックが各楽章を構成する最小単位となったということではない。楽章を制作するための各作業や楽章を構成している単位が、この基準の単位よりも時間的、機能的に広い範囲を扱っているのか、あるいは、この基準の中に内包されるものを扱っているのかという観点からとらえられるということである。

そして、このような観点を持つことができることによって、楽曲の構造を考える際に、ヒエラルキー構造として捉えたり、トップダウンやボトムアップなどの既存の作曲の方法論に当てはめたりすることが可能になった。

楽曲の構造をヒエラルキーとして捉えるというのは、楽曲を構成する部分構造の組織を、時間軸上の長さや楽曲における機能という点から細かく分類し、楽曲の全体構造(もっとも大きな時間の構成)をトップ、もっとも小さな構造(音符など)をボトムと考えることである。そして、これは作品制作の各工程での作業の対象を明確にする考え方である。このような捕らえ方が制作時に可能になったことで、作業の対象の作品における位置づけがはっきりしたといえる。

## 7. おわりに

今回、筆者の作品「Four tears」の 2 つの楽章とこれらを制作するために用いた CMask を題材として、作曲のためのツールを利用する際の方法について、検討を試みた。

このような検討を行ったのは、作り手として、作品について総括する必要があると感じたことと、現在の筆者が持っている、コンピュータの利用した音楽制作において、音楽の構造や音楽制作のプロセスをどのように考えるべきなのかという疑問に対するひとつの取り掛かりとなることを期待してのことである。

しかし、まだ勉強不足、検討不足の部分が多く、本稿は、十分にまとまっているとは言いがたい部分が多い。皆様からのご指導を賜れば幸いです。

## 謝辞

本稿の執筆に当たって、東京大学の中村文隆先生に、多岐にわたってご指導いただきました。この場をお借りしてお礼を申し上げます。ありがとうございました。

#### 参考文献

- [1] Richard Boulanger 他:「The Csound Book」、The MIT Press、2000
- [2] Curtis Roads・青柳他訳:「コンピュータ音楽 歴史・テクノロジー・アート」、東京電機大学出版、2001年、pp645-661, pp750-751
- [3] <http://www.kgw.tu-berlin.de/~abart/>
- [4] Robert Train Adams:「Electronic Music Composition for Beginners Second Edition」、MacGraw-Hill、1986年、pp200-201
- [5] Eduardo Reck Miranda:「Composing Music with Computers」、Focal Press、2001、pp9-10