

音合成システム「おっきんしゃい」のデータ構造と GUI

小坂直敏

NTT コミュニケーション科学基礎研究所

〒243-0198 神奈川県厚木市森の里若宮 3-1

Osaka@brl.ntt.co.jp

あらまし Windows上の視覚的・操作的環境上で音合成、加工、演奏の機能を持つ音合成システム「おっきんしゃい」の検討を進めている。本稿では、その中で用いる音のデータ構造とGUI(グラフィカル・ユーザ・インタフェース)を新たに検討した結果について述べる。従来の音楽生成ソフトウェアは、譜面情報を主に扱うシーケンサと音合成ソフトウェアとが異なる概念のものとして発達してきた。

「おっきんしゃい」では、音楽から音までを一つの音オブジェクトという概念で統一的に扱う。また、単一の音波形を有す単音オブジェクトには演算履歴が内包し、複合音オブジェクトには音オブジェクト間の演算が表現されている。このような新しいシステムを用いた音楽制作と発表もすでに行われ、ひとまずの成功を修めている。

キーワード 音合成, GUI, 音オブジェクト, メルティング

Sound Synthesis system “*O^tkinshi*”: its data structure and Graphical User Interface (GUI)

Naotoshi Osaka

NTT Communication Science Laboratories

3-1 Wakamiya, Morinosato, Atsugi-shi, Kanagawa, 243-0198

Osaka@brl.ntt.co.jp

Abstract We are studying a sound synthesis system “*O^tkinshi*” which has functions of sound generation, modification and performance under the visual operation environment. This paper describes the new concept of sound data structure and GUI (Graphical User Interface) needed for the system. Previous music generation software has been developing, distinguishing a sequencer which deals with common music notation and sound synthesis software. In *O^tkinshi* “sound object” covers universally all the concepts from music to sound. Moreover, Operation history is implemented in a single sound object, in which only a single sound waveform is defined, and compound sound object expresses operations of inter-sound objects. Music creation and performance were successfully done using the new system.

Key words Sound synthesis, GUI, Sound object, Melting

1. はじめに

コンピュータ音楽作品の創作やマルチメディアのコンテンツのための音合成ツールとしてこれまでいくつものハードウェア/ソフトウェアシステムが知られている(1),(2).

これらは譜面情報に基づく音楽生成と、音合成を中心にしたものとの二つに大きく分類できる。前者はシーケンサと呼ばれ、譜面上の操作が一般的である。音源はMIDI制御による発音が主で、これに音合成された音源などが加わる。

一方、音合成ソフトは譜面にとらわれず、音波形を合成することが目的である。扱った作品により、譜面情報(ピッチ、音長など)が重要でない場合も多い。こうした状況で、音合成ソフトは必ずしも譜面情報とのリンクを目指すのではなく、MAX(1)のように音楽実演の中で、一般の楽器と同様に実時間音合成を操作性よく実現させることが一つのねらいである。もう一つのねらいは、より緻密な音合成を目的としたオフライン音合成システムである。「おっきんしゃい」は後者の枠組みである。これまで音楽生成と音合成のシステムは、上述のように別の体系であり、創作に使用するためには操作上不便であった。本報告では、この不便さを解消するよう、音楽から音までを統一的に扱うための音のデータ構造とそのGUIについて述べる。

2. システムの構成

現行システムは(3)に詳しく述べてある。その概要を図1に示す。システムは、大きく分けて音合成部と演奏部に分かれる。音合成部は汎用機能と本システム固有の特徴的なものがある。汎用機能として、一般的な波形エディタ機能と、正弦波モデルに基づく音合成および編集/変形機能がある。システム固有の機能として、正弦波モデルに基づくビブラート制御(ビブラートの付与および除去)(4)と音色モーフィングがある。音色モーフィングは信号モデルによる方法(5)と物理モデ

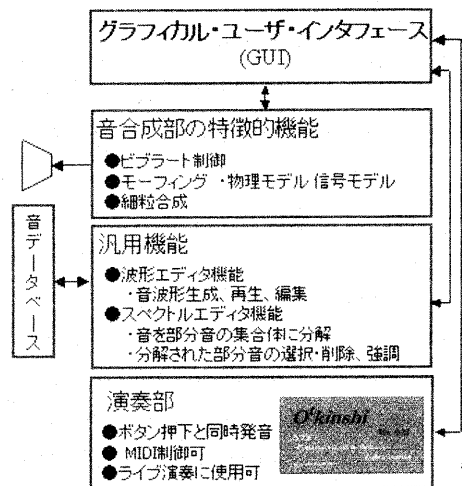
ルによる方法(6)との二手法を実装している。

今回新たに細粒合成(7)を実装した。ここでは、原波形に窓かけを行いオーバーラップアッドする分析合成に基づく手法で、窓のサイズ、および再合成時での時間軸での布置、重なり具合をパラメータに加えたものを実装した。

演奏部は得られた音オブジェクトのアイコンをパネル上に順次配置したもので、これの押下により音が提示できる。ここのシーケンスの複雑な組み合わせは下のレイヤで実現しているため、最上位である演奏部では、その機能を付与していない。

楽曲の創作時にユーザは、以下の観点でシステムを使用する。

- 1) 時間の経過とともに起こる目まぐるしい音色の変化の設計と実現。
- 2) 音楽の大局的な流れから、一つの楽句の構成、あるいは一秒程度の音素材、数ミリ秒の音色まで、テンポラルレートと音楽的階層の異なる領域での音合成の設計とその実現。
- 3) いくつもの音色の重ね合わせとその音量、タイミングの調整。



Windows PC
図1 システムの構成

このように、ユーザは、音のミクロのレベルから音楽というマクロのレベルまで、また併せて時間的経緯についても思考の対象とレベルが往来する。「おっきんしゃい」では、音オブジェクトという概念で音楽から音までを統一したデータ構造を持たせることにより、創作時に音楽のトップレベルからそのミクロのレベルまでの操作の行き来を思考と同期して追従できるよう工夫したものである。

3. データ構造と GUI

本システムで扱うデータ構造を表1に示す。それらは、1)音オブジェクト(Sound object), 2)演算オブジェクト(Operation object), 3)その他の音の属性を表す属性オブジェクト群に分類できる。また、この他に音オブジェクトを指し示す音オブジェクトのポインタがある。

演算オブジェクトは各種の信号処理演算を施すためのオブジェクトである。属性オブジェクトは、音オブジェクトと異なり、音の属性の一部を表現するもので、これからは原音の復元はできない。

3.1 音オブジェクトの階層的表現

音オブジェクトは複合音オブジェクトと単音オブジェクトとに大別でき、音を知覚的に完全に復元できるデータ構造として設けられている。これは階層的かつ入れ子構造となっている。音オブジェクトが階層的な表現される様子を図2に示す。最下層では単音オブジェクトが一つの波形デ

ータを定義し、これらが階層的に組み合わさったものを複合音オブジェクトとする。中間層では楽句を表し、最上層では一つの楽曲を表す。

音オブジェクトに限らず、全オブジェクトはその表示に二つの階層(レイヤ;Layer)を設けてある。上位のレイヤ(レイヤ 0;(Layer 0))では、そのオブジェクトを象徴するアイコン付きのボタンで表される。これらはオブジェクトに応じて音アイコン、演算アイコン、ピッチアイコン、ビブラートアイコンなどと呼ぶ。下位のレイヤ(レイヤ 1;(Layer 1))では当該オブジェクトの詳細表示を行う。レイヤ 0 へのアクションはマウス操作を簡易にするために設けられ、それぞれのオブジェクトの中で、最も本質的な、あるいは頻度が高く使用される機能である。音オブジェクトでは、パイロットプレイと呼び、シングルクリックにより音オブジェクトの冒頭数秒間の音が再生される。レイヤ 0 から 1 はダブルクリックで、また逆方向へはクローズボタンによって移行できる。

レイヤ 1 では、複合音オブジェクトと単音オブジェクトとでは表示内容がまったく異なる。

複合音オブジェクト

同オブジェクトのレイヤ 1 には文字通り複数の音オブジェクトが参照されている。ここには複合

表1 システムで用いるオブジェクト

種類	内容
音オブジェクト	単音オブジェクト
	複合音オブジェクト
演算オブジェクト	レベル変換, ピッチ変換, フィルタリング他
属性オブジェクト	ピッチ, スペクトル包絡, ビブラート, 正弦波モデル表現パラメータなど

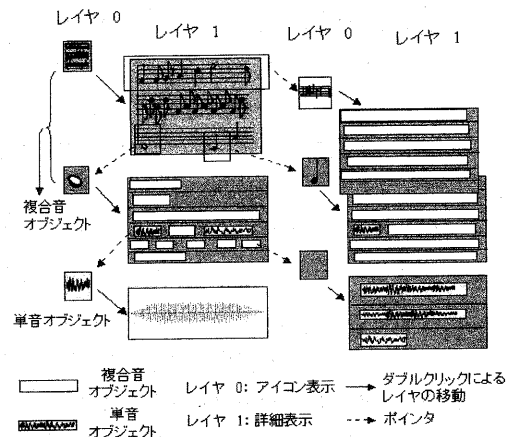


図2 音オブジェクトの階層的表現

音オブジェクトを入れ子として、あるいは、単音オブジェクトを終端子として指し示すことができる。ここで指し示す下層の音オブジェクトは図2に示すとおり、再び同様に二つのレイヤを有す。こうして、どの音オブジェクトも二つのレイヤを相対的に持つ。上下に隣接する各音オブジェクトはレイヤ1とレイヤ0で階層的な関係が定義されるため、図の矢印に添ってレイヤ数を加算すれば、当該オブジェクトが全体オブジェクトの中で何番目の階層かが知られる。

単音オブジェクト

音オブジェクトのレイヤ1はモニタ部(Monitor)と演算履歴部からなる。モニタ部では、波形/スペクトル表示(Wave/Spectrum monitor)、音の録音/再生(Record/playback)などからなり、演算履歴部では、家系図のように生成のルートからのユーザの演算実行履歴が示される。

4. 音オブジェクトと演算の表現

音オブジェクトと演算との関係を以下のように整理した。

複合音オブジェクトでは、音オブジェクト間の演算を定義し、その結果得られる音をオブジェクトとする。演算としては、信号データと窓関数データとのコンボリューションなども考えられるが、現在は、開始時点、レベルなどを個々の音オブジェクト毎に記述した加算機能のみ実装している。これは、いわゆるミキシング機能である。

一方、単音オブジェクトは、生成(あるいはファイルからの波形読み込み)後に波形がどのような変形を受けたかの履歴を記述した。

図3は二つの音オブジェクトのレイヤ1を表示したものである。a)は単音オブジェクトで、演算アイコン列として履歴が残る。この履歴はスクリプトとしてユーザにも提供される。ユーザはこれを編集して再合成(再計算)することができる。これにより、MAXのパッチと同様に視覚的なプログラミング環境の機能(1)も有すといえる。

このプログラムを実行するためには、本システムとオブジェクト指向の環境とを並行して立ち上げておき、この環境内で編集されたコマンドファイルを実行することにより「おっきんしゃい」の音オブジェクトが作成される。

この時履歴の表示はパイプラインの表示になる。現在はオブジェクト指向環境として Ruby(8)を用いている。Rubyは言語が単純であることのほか、インターネットからダウンロード可能で実装が容易であること、および、MS-DOS上で動作する点、パソコンで用いるのに使用が手軽である点などが特徴である。

図3b)は複合音オブジェクトのレイヤ1表示で、冒頭の部分がずれている音同士を加算した例である。スクリプトにはデータが記述されている。複合音オブジェクトには演算履歴を設けていない。これは、複数のさまざまな履歴を持つ音オブジェクト間の演算の結果として合成された音であるため、パイプライン的な履歴表示とはなりえず、直前の演算に関わった音オブジェクトを表示することが最も単純な表示である。単音オブジェクトと同様な履歴をみるためには、図2に示すように、全ての音オブジェクトをダブルクリックをし続けて単音オブジェクトのみに展開することが演算履歴を表示したことになる。

5. 表示モードの切り替え

ユーザは、創作の工程によっては、波形、あるいはスペクトル以外の他の属性で音を見ながら操作したいことが多い。具体的には音高、音長などの一般的な譜面情報の方が操作しやすい場合も多い。そこで、音オブジェクトを波形以外の属性オブジェクトで表示できるようにした。これは、データ構造を音オブジェクトのまま変えないで、表示だけを任意の属性オブジェクトにするものである。

この表示モードの切り替えのため、メルティング(Melting)と呼ぶ新しいマウス操作を考案した。

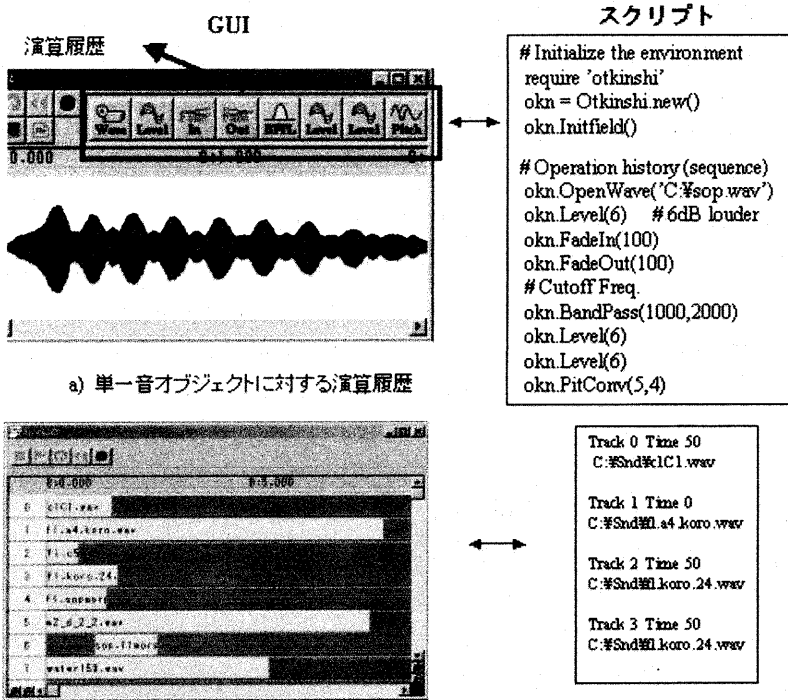


図3 音オブジェクトと演算の表現

この操作は、ハンダに熱いこてを暫く当てるとやがてハンダが溶ける様子を擬似したものである。これで、以下の二つの機能を実現する。

- メルティング機能によるモードの切り替え (図4 a)。アイコンが単独で存在する場合、すなわち、マウスでポイントしているアイコンが他のアイコンに触れていない場合。このときは一定時間以上マウスを押下したままにするとこのアイコンが溶解して別のモードになる。ここでは、音オブジェクトに対するメルティング機能として、波形表示モードから譜面表示モードへの、あるいはその逆方向へのモード変換がある。
- メルティングによる二つの属性の融合(図

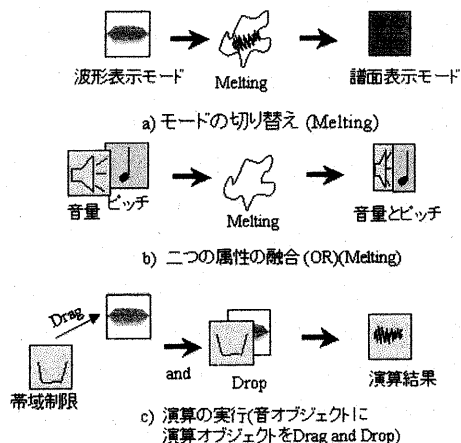


図4 マウスによる操作性の向上

4b). アイコンが二つ重なっているとき、一方にマウスを一定時間押し続けるとメルティングが起り、二つの属性が融合した属性が形成される。これらは、マウス操作の新たな使用方法として先例になるであろう。

このほかに単純化した操作として、レイヤ0間での演算の実行がある。これは、音オブジェクトに対し、演算オブジェクトをドラッグした後マウスから手を手放す(Drag and Drop)ことにより実行される。

6. まとめ

Windows上の音合成、加工、演奏システム「おつきんしゃい」について、上位の音楽的操作から下位の音合成の操作までを、ユーザの思考のレベルに応じて迅速に行き来できるためのデータ構造とGUIを検討した。

音オブジェクト、演算オブジェクトを主たるオブジェクトとするが、特に音オブジェクトに階層性を持たせ、複合音オブジェクトと単音オブジェクトとを定義し、複合音オブジェクトは下層の複合音オブジェクトと単音オブジェクトを要素に持ち、これらの演算結果として表現される。

単音オブジェクトはデータであるが、その中に発生から現段階まで自身の受けた演算履歴が記録されており、これが簡易なプログラム言語となっていることを述べた。また、複合音オブジェクトは音オブジェクト間の演算を表現するもので、現在は音オブジェクトの加算(ミキシング)機能のみを実装している。これらはスクリプトで表現されており、編集可能である。

本システムはこれまでいくつかの実演を経てきた。今回の新たなバージョンでは、本年3月に新たな実演の機会(P-1)を得て、2作品をステージで発表し、システムのトラブルもなく実演に成功した。今後は大学での講義での使用など、教育的な場でシステムのモニタを依頼し、改善していく予定である。

謝辞

この研究を行う機会を与えていただいた萩田紀博メディア情報研究部長、および研究を進めるにあたり日頃議論していただくメディア表現グループの諸氏に感謝します。

参考文献

- (1) M. Puckette, "Combining event and signal processing in the MAX graphical programming environment," *Computer Music Journal*, Vol 15. No. 3, pp. 68-77, 1991.
- (2) K. J. Hebel and C. Scaletti, "The software architecture of the Kyma System," *Proc. of Int. Computer Music Conf. 95*, pp. 164-167, Tokyo, 1993.
- (3) 小坂直敏, 榊原健一, 引地孝文, "Windows上の音合成システム「おつきいんしゃい」の構築," *信学論(D)*, 2001.6 掲載予定
- (4) K.-I. Sakakibara and N. Osaka, "Vibrato control using a sinusoidal model," *J. Acoust. Soc. Jpn.*, 25. 2000.
- (5) N. Osaka, "Timbre interpolation of sounds using a sinusoidal model," *Proc. of Int. Computer Music Conf. 95*, pp. 408-411, Banff, 1995.
- (6) T. Hikichi and N. Osaka, "Sound timbre interpolation based on physical modeling," *J. Acoust. Soc. Jpn.*, Feb. 2001.
- (7) 青柳龍也, 小坂直敏, 平田圭二, 堀内靖雄, "コンピュータ音楽—歴史・テクノロジー・アート—," 東京電機大学出版局, 2001 1.
- (8) まつもとゆきひろ, "オブジェクト思考スクリプト言語Ruby," *SOFTWARE SCIENCE シリーズ*, アスキー出版局, 1999. 10.

演奏

- (P-1) NTT コミュニケーション科学研究所主催,
NTT コンピュータ音楽コンサートII, 東京
津田ホール, 3/8/2001.