

ピアノ響板シミュレータのコンポーネント化の試み

A case study of Component for Piano Soundboard Simulator

能崎友博¹⁾, 梅谷征雄²⁾

Tomohiro NOZAKI and Yukio UMETANI

1) 静岡大学 情報学研究科情報学専攻 (〒432-8014 浜松市城北3-5-1, cs9068@cs.inf.shizuoka.ac.jp)

2) 工博 静岡大学教授 情報学部情報科学科 (〒432-8014 浜松市城北3-5-1, umetani@cs.inf.shizuoka.ac.jp)

A component oriented visual / interactive simulator is a PSE which can perform the simulation of physical phenomena based on the connection graph of physical components. We implemented a new piano soundboard component using FEM (Finite Element Method) by converting an existing soundboard simulator to a ATL COM (Component Object Model) component. The process of conversion is described here.

Key Words : Piano soundboard, simulator, FEM, ATL COM

1. はじめに

物理シミュレーションにおいてそのモデルの忠実度を上げるためにはパラメータの感度解析や実測値との対比が重要な役割を占める。このような操作を容易にするためには、応答性と対話性に優れたシミュレーション環境が必要と考える。[1]

我々は物理コンポーネントを組み合わせるシミュレーション対象を構築する「コンポーネント指向可視化対話型シミュレータ」の研究を進めているが、そこでは物理コンポーネントの品揃えが成功のカギとなる。既に開発済のピアノシミュレーション用の物理コンポーネント群（弦、駒、差分法響板）に新たに有限要素法によるピアノ響板コンポーネントを追加して更に精度の高いシミュレーション環境を構築するために、独立したアプリケーションである「可視化対話型平板シミュレータ」内の有限要素法を使った振動解析部をコンポーネントに仕立て直す作業を行った。これを通してコンポーネント化に必要な工程と工数を見積もった。[2][3][4][5]

2. 有限要素法ピアノ響板コンポーネント作成の準備

(1) 手順

コンポーネント作成は以下の手順で行う。

1. コンポーネント指向可視化対話型シミュレータの構造を解析する。
2. 差分法響板コンポーネントのプログラム構造を解析する。
3. 可視化対話型平板シミュレータのプログラム構造を解析する。
4. 機能的強度が強くなるよう、平板シミュレータ

のモジュールの独立性を高めて、モジュールを独立で使うことが出来るようにする。

5. インターフェース関数の機能に準じたモジュールを取り出し、インターフェース関数として実装する。
6. 残ったモジュールのうち響板コンポーネントに必要なモジュールとそうでないモジュールとに分別する。
7. インターフェース関数の機能にそぐわないモジュールはインターフェース外部のメンバ関数として実装する。

(2) コンポーネント指向可視化対話型シミュレータ

様々な物理モデルを1つのコンポーネントとして作成して用意し、それらのインスタンスであるオブジェクト単体で、またはオブジェクト同士を繋ぎ合わせることでシミュレータが作成できるようなシミュレーション環境を開発することを目的として作成されたシミュレータである。

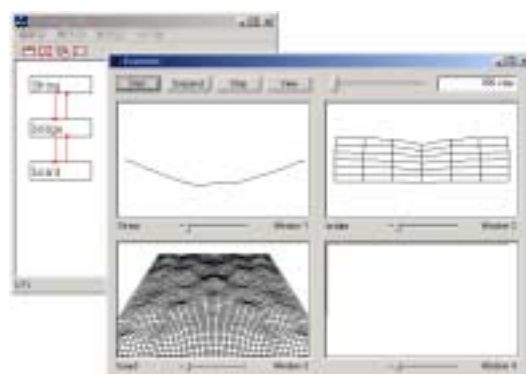


図1 コンポーネント指向可視化対話型シミュレータ

図1はコンポーネント指向可視化対話型シミュレータを使って弦・駒・響板コンポーネントの3つを接続し実行している画面である。

それぞれの物理コンポーネントには共通のインターフェース関数(関数群)を持たせてあり、それぞれの物理コンポーネントにアクセスする時にはそのインターフェースを通してアクセスを行うことになっている。

(3) 差分法ピアノ響板コンポーネント

異方性平板の曲げ振動方程式を差分近似し、計算式を得る。時間・空間について分割し、それを中央差分を用いて近似し、前進解法の陽公式で計算式を求める。その計算式をプログラム化し、実装したものが差分法ピアノ響板コンポーネントである。このコンポーネントは、コンポーネント指向可視化対話型シミュレータの規格に準じたインターフェースを持っている。

解析した結果、パラメータ変更に関する部分・計算に関する部分のインターフェースを変更すれば、他の部分のインターフェースは有限要素法響板に流用できると判断した。よって、有限要素法ピアノ響板コンポーネントは差分法ピアノ響板コンポーネントをベースに作成する。

(4) 可視化対話型平板シミュレータ

このシミュレータは物理モデルとしてピアノの響板にあたる平板の曲げ振動を取り上げ、そのモデルの空間形状・メッシュサイズ・パラメータ値などの変更の影響を即座に計算して表示することにより、モデルの調整を効率よく行えるものである。(図2)

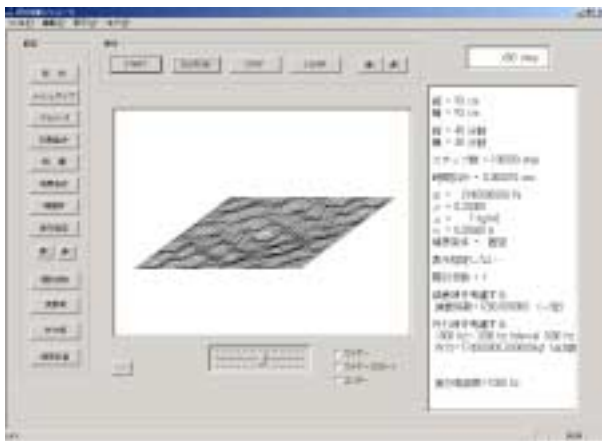


図2 可視化対話型平板シミュレータ

プログラム構造を解析した結果、計算実行部分の中心となる関数はOnstartという関数であることが判明した。この中でシミュレーションを行っている部分のプログラム部は、多くの条件分岐を行っている。この条件分岐は、減衰項があるかないか、外力を与えるか否か、境界条件は何であるか、という判定条件によって分岐し、それらの条件は全てパラメータによって与えられるものであった。

今回コンポーネント化をするにあたって、響板コンポ

ーネントには減衰項は必要であること、外力は駒から必ず与えられなければならないことを踏まえ、減衰項と外力に関するパラメータ変更部の実装は行わない。減衰項があり、外力を与えるという条件のもとで分岐したプログラムコードを実装する。ただし、境界条件のパラメータは他コンポーネントから必ず与えられるわけではないので、ユーザが設定できるようにプログラムを作成する。

3. 有限要素法ピアノ響板コンポーネントの作成

(1) 改変コード量

コンポーネントを作成するにあたり、インターフェースの変更、メンバ関数の作成・追加、データメンバの作成・追加を行った。(表1)

表1 コンポーネント化した際に新規作成・追加変更したコード量

	総数	新規	追加	変更	改変コード量
インターフェース	26	0	0	8	約 230 行
データメンバ	133	5	42	0	約 50 行
メンバ関数	26	6	10	4	約 1200 行

可視化対話型平板シミュレータで使用していたデータメンバ・メンバ関数の流用が多いため「追加」の欄の値が大きくなっている。当然だが、インターフェースは規格に基づいて作成されているため「新規」は1個もない。

(2) インターフェース関数

有限要素法ピアノ響板コンポーネントは規格化されたインターフェース関数(表2)を持つ。このインターフェース関数を通じて、クライアントとのデータ通信を行う。

表2 インターフェース関数のリスト(●は差分法コンポーネントと比較してプログラムを変更したインターフェース関数)

```
STDMETHOD(Init());
●STDMETHOD(InitPara());
●STDMETHOD(GetPara)(double *data);
●STDMETHOD(SetPara)(double *data);
STDMETHOD(RCGetSmp)(double *data);
STDMETHOD(RCSetSmp)(double data);
STDMETHOD(GetFqA)(double *f, double *a);
STDMETHOD(SetFqA)(double f, double a);
●STDMETHOD(InitCalc());
●STDMETHOD(Calculation());
●STDMETHOD(Calculation2());
STDMETHOD(InitCommuni());
STDMETHOD(InputPort0());
STDMETHOD(InputPort1)(double *data);
```

```

STDMETHOD(InputPort2)(double *data);
STDMETHOD(OutputPort1)(double *data);
STDMETHOD(OutputPort2)(double *data);
STDMETHOD(CheckView());
STDMETHOD(DeleteView());
STDMETHOD(InitView)(HWND hWnd, int w, int h);
STDMETHOD(ReInitView());
●STDMETHOD(MakeBitmap());
STDMETHOD(SendBitmap());
STDMETHOD(SetWH)(int w, int h);
STDMETHOD(SetAmp)(int i);
STDMETHOD(GetPV)(double *data);

```

(3) 変更したインターフェース関数の説明

差分法響板コンポーネントのインターフェース関数の中で、有限要素法響板を作成するにあたり変更したインターフェース関数について説明する。

●STDMETHOD(InitPara)();

各種パラメータの初期値を設定する。SetPara(パラメータ変更用インターフェース関数)で何も値が設定されない場合は、この初期値を用いてシミュレーションが実行される。シミュレーション実行前にユーザから与えられたパラメータを格納するデータメンバはここで初期値を設定、つまり初期化される。

有限要素法ピアノ響板に必要なパラメータ(響板形状の選択・ポアソン比)が追加された。

●STDMETHOD(GetPara)(double *data);

クライアントがパラメータ値を参照した場合に用いられ、1次元配列dataを使って各データ値を渡す。

InitParaと同じく新しいパラメータが追加された。

●STDMETHOD(SetPara)(double *data);

クライアントがパラメータ値を設定する場合に用いられる。

InitParaと同じく新しいパラメータが追加された。

●STDMETHOD(InitCalc)();

計算用のデータメンバを初期化するためのインターフェース関数。コンポーネントで使用する配列の初期化もここで行われる。

有限要素法の計算に必要である2つのメンバ関数 numbering() と matrix() の呼び出しがここで行われるようになった。

●STDMETHOD(Calculation)();

曲げ振動の計算の際に用いられ、このコンポーネントでもっとも重要な部分である。境界条件にしたがったメンバ関数を呼び出して計算を実行する。

メンバ関数 phase_gen, setw の呼び出しを行って、減衰項・外力の影響を受けるようにした。

●STDMETHOD(Calculation2)();

コンポーネント単体で実行する際に呼び出されるもので、役割は Calculation と変わらない。Calculation と同様の

変更を行った。

●STDMETHOD(MakeBitmap)();

表示用に用意された構造体 point に実行結果と与えられたパラメータから計算された節点の座標値を入れ、節点と節点を直線で繋ぐことで各要素を描き、響板の形状をビットマップで描画する。

(3) 追加したメンバ関数とインターフェース関数

a) パラメータ変更部

パラメータ変更部では、3種類のインターフェース関数のうち、パラメータ設定のインターフェース関数 SetPara 内でのみメンバ関数が使われる(図3)。メンバ関数 FileOpen は実形状響板データを読み込む際に使用される関数である。そのため、長方形響板のシミュレーションの際には使用されない。

実形状響板を使うというパラメータが与えられた場合、まず GetFilePath を使って実形状データファイルへのパスを獲得する。その後、3種類ある実形状データファイルをそれぞれ専用のメンバ関数3種類を使って開き、中のデータをデータメンバにストアする。

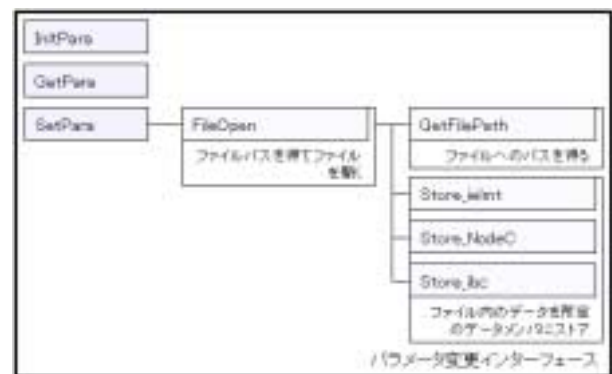


図3 パラメータ変更メンバ関数接続図

b) 計算実行部

計算実行部には、数多くのメンバ関数が実装されている(図4)。有限要素法による離散化を行った計算式を計算機上で実現するためには数多くのコードが必要である。

InitCalc インターフェース関数では、主に計算に関するデータメンバの初期化に用いるメンバ関数が呼び出される。numbering では境界上節点番号や要素が持つ節点番号の指定を行う。ただしこの作業は実形状響板の場合、既にファイル内にデータがあるため必要ない。matrix では全体係数マトリックスを計算する。

Calculation, Calculation2 インターフェース関数では境界条件にしたがって calc_board_fix, calc_board_support, calc_board_free のいずれかを呼び出して計算を行う。計算が終わったら phase_gen で駒コンポーネントからの外力を受け取って結果に反映させる。また、駒コンポーネントへ与える変位値の送信を getzu にて行う。



図4 計算関係データメンバ接続関係図

4. 有限要素法ピアノ響板コンポーネントの実装

作成した有限要素法ピアノ響板コンポーネント (board_FEM.dll) をコンポーネント指向可視化対話型シミュレータに実装してコンポーネントの評価を行う。

(1) 実装方法

有限要素法ピアノ響板コンポーネントをコンポーネント指向可視化対話型シミュレータに実装するためにはクライアントのコードを改変しなければならない。

改変するにあたって、差分法響板コンポーネント (board_exp.dll) やその他のコンポーネントがクライアント上でどのようにして呼び出されているかを調べる。調べた結果を元にして響板コンポーネントを実装できるようにコードを改変する。

(2) 評価

有限要素法ピアノ響板コンポーネントを単体で実行しシミュレーションを行う。また、可視化対話型平板シミュレータについてもシミュレーションを行う。このとき、与えるパラメータ値を全て同じにして、任意の5つの節点での各ステップでの変位値を調べ、コンポーネント化の正当性を確認することにした。両プログラム中の計算を行うコードの後に、その結果をファイルに出力するようなコードを追加する。

その結果、双方とも各節点それぞれのステップでほぼ同じ変位値をとった。よって、コンポーネント化は正しく行われたと言える。

5. おわりに

本研究では有限要素法ピアノ響板コンポーネントを開発した。しかし、開発したコンポーネントは使用するうちに様々な問題が出てきた。

コンポーネントのパラメータのうち、分割数は40×40分割を初期値として設定してあり、この分割数ならばプログラムは正しく動作する。しかし43×43分割のような、

少し多い分割数に設定するとエラーが起きる。これは、節点を保存するデータメンバが1700節点分の領域しか確保していないために起こるエラーである。従ってこのデータメンバのサイズを大きくすればエラーは発生しないように思われる。しかし、このように1700節点分の領域をとるデータメンバは他にも多く作られており、また1700×1700の2次元配列もいくつか作られている。これらのデータメンバは静的に領域を確保しているため、最初からあまりに大きな領域を確保すると、プログラムを起動しただけでメモリ不足に陥りかねない。よって、これらのメモリ確保を動的に行うようにすべきであろう。

また、ANSYSから出力した実形状の響板データを読み込んでシミュレーションを行う機能も付加する予定である。現在のところファイルの読込部と実行結果の表示部が完成している (図5)。

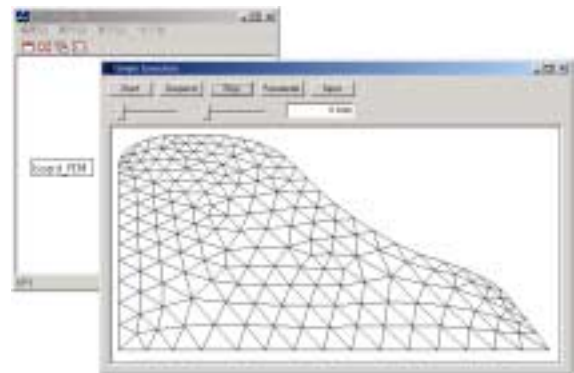


図5 実形状響板を読み込んだ際のシミュレータ画面

コンポーネント化を行うための工程 (2.(1)) を補助するようなウィザードがあれば、簡単にコンポーネントを作成できる。しかしそのためには、アプリケーション内のプログラム構造を分析して関数の役割を判別する機能やインターフェース関数の機能に準じたモジュールを抽出する機能などを完成させなければならない、実現には多くの時間を要するであろう。

参考文献

- 1) 梅谷征雄：PSE 可視化インターフェース，第一回問題解決ワークショップ論文集，pp25-30，1998
- 2) 吹原教弘，梅谷征雄：可視化対話型物理シミュレータの試作と応答性の評価，計算工学講演会論文集，Vol.4，No.1，pp225-228，1999
- 3) 竹下明良，梅谷征雄：有限要素法可視化対話シミュレータの試作，計算工学講演会論文集，Vol.5，No.2，pp543-546，2000
- 4) 菊地文雄：有限要素法概説 [新訂版] -理工学における基礎と応用-，株式会社サイエンス社，1999
- 5) Richard Grimes, Alex Stockton, George Reily, Julian Templeman 著 田中正造 監訳：ATL COM プログラミング，翔泳社，1999