協調的分散オブジェクト技術によるWebベースの有限要素解析システム

A Web-based Finite Element Analysis System Using a Collaborative Distributed Object Technology

室蘭工業大学建設システム工学科　正会員　矢吹　信喜　(Nobuyoshi Yabuki)
室蘭工業大学大学院工学研究科建設システム工学専攻　学生員　○岩崎　充乗　(Mitsunori Iwasaki)
室蘭工業大学建設システム工学科　学生員　照井　陽祐　(Yousuke Terui)
Department of Civil and Environmental Engineering, Stanford University　Kincho Henry Law

1．はじめに

1980 年頃までは有限要素解析（FEM）コードは，各企業や機関における自主開発が基本であったが，その後，最新のアルゴリズムや要素技術を取り入れ，大規模な FEM コードを開発・メンテナンスしていく事がだんだんと困難になっていった．それと同時に米国では，商用 FEM コードが台頭し，信頼性が向上すると共に，我が国でも商用コードの購入・使用が一般的になった．

その結果，FEM 解析が容易に実施できるようになったが，自ら必要とするが商用コードがサポートしていない要素の開発が難しくなり，必要な時にすぐに対応できないといった問題が発生してきた．さらに，商用コードは，大規模なソフトの開発やメンテナンスを行う事から高価格になり，こうした高価格な商用 FEM パッケージを揃えられるのは一部の大企業等に限られ，中小企業とのギャップが顕在化してきたと思われる．

大規模 FEM コードの開発・メンテナンスコストを大幅に低減するためには，従来のような Fortran77 に代表される手続き型言語からオブジェクト指向プログラミング言語に変更し，コード（オブジェクト）の中身を見なくても容易に再利用できるような環境を構築していく必要があると考えられる．

さらに，Linux の開発のように多数の開発協力者が最新の要素を取り入れたコード(オブジェクト)を開発し，協調的に FEM コードの高度化を図っていく事が望まれる．実際，こうした実行環境に関する研究[1]がなされている．

また，FEM コードをサーバやサーバと連動する他のコンピュータなどに分散して配置することにより負荷が１台に集中するのを防ぎ，ユーザはインターネット上のクライアントとしてサーバへアクセスして，適切な課金等の方法により FEM 解析が可能なシステムの構築をしていく必要があると考えられる．

本研究では，オブジェクト指向プログラミング言語 Java により，２次元静的弾性有限要素解析プログラムを開発し，ORB（Object Request Broker）技術によって，WWW 上で分散処理が可能なプロトタイプシステムを開発した．

2．オブジェクト指向による２次元静的弾性 FEM コード

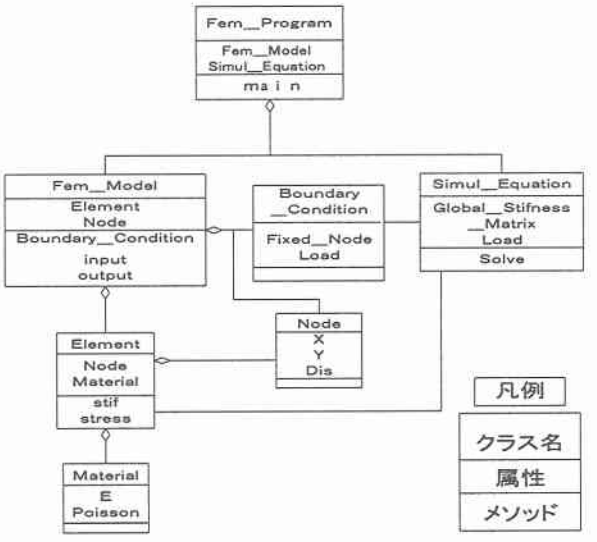
ほとんどの FEM コードは，構造化プログラミング手法を用いて作成されており，サブルーチンのライブラリ化によって，開発やメンテナンスを容易にする努力がなされている．オブジェクト指向プログラミングによるアプローチは，構造化手法が目指すものをさらに前進させたものであり，アルゴリズムのみならず，データも一つの塊（オブジェクト）に封じ込め，プログラムはオブジェクト同士の相互作用（メッセージパッシング）により動作するものである．

本研究では文献[2),3)]を参考にしながら２次元静的弾性有限要素解析プログラムを Java により開発した．本コードのクラス図を図－１に示す．

尚，本プログラムは２次元静的弾性有限要素解析を実際に実行する事は可能であるが，研究を目的としたものである事から，要素には３角形要素，連立１次方程式の解法としては Gauss-Jordan の消去法を用いており，機能的にはベーシックである．しかし，オブジェクト指向パラダイムを用いているので，機能を拡張していく事は容易だと考えられる．

3．分散オブジェクト指向による WebFEM

ネットワーク上には数多くのコンピュータが分散しており，各コンピュータにはアプリケーションソフトウェアがある．これらのアプリケーションソフトを連動させるには各アプリケーションソフトをオブジェクトとして封じ込めて，必要な時にオブジェクト同士で動作をリクエストする図－２に示すようなフレームワーク，すなわち ORB が必要である．そこで，1989 年にはオブジェクト環境開発を目的として OMG（Object Management Group）が設立され，1991 年には CORBA（Common Object Request Broker）1.1 が制定された．



図－１　FEM コードのクラス図

これにより，異なった言語で書かれた分散オブジェクトを利用した大規模アプリケーションソフトの開発が可能となったが，OMG の仕様による IDL（Interface Definition Language）によるプログラム開発は必ずしも容易とは言えない．

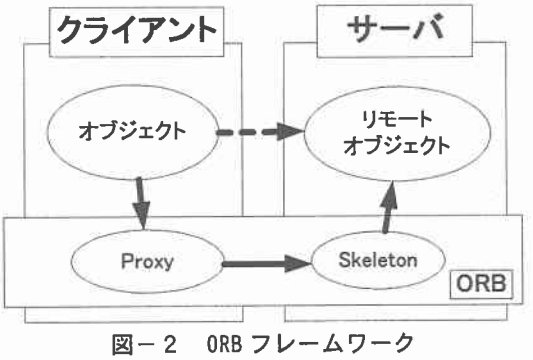
そこで，Java 用分散オブジェクトソフトウェアである HORB[4]（Hirano Object Request Broker）が通商産業省工業技術院電子総合研究所で開発された．HORB を用いると，難解な IDL によるプログラミング作業が必要なく，容易に分散オブジェクトプログラムが開発できることから，本研究では HORB を使用することにした．尚，各アプリケーションのプログラミング言語は Java に限定されず，Fortran 等で作成することも可能である．以下，本研究で開発した，ユーザが WWW 上で使用可能な WebFEM について記す．

4．WebFEM のシステム構築について

本システムの構築にあたっては，アプリケーションソフトウェアの作成には Java 言語，ユーザインターフェースの作成には HTML 及び JavaApplet，分散オブジェクトプログラムの作成には HORB を使用した．開発環境を表－1 に示し，システムモデル図を図－3 に示す．

まず，ユーザは Web を経由してサーバ上の Web ページへアクセスする．そこで，あらかじめ作成しておいた FEM 解析用の入力データを JavaApplet を通じて送信する．すると，その送られてきた解析データを元にサーバ側では他のサーバへ処理を分散して解析を行い，サーバは解析結果に基づいて結果を表示するための Web ページを作成する．ユーザは再び Web を経由して，その Web ページへアクセスする事で解析結果を知る事が出来る．

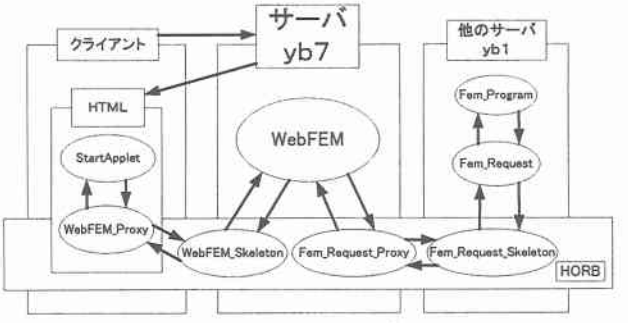
このような分散処理を可能にするのが HORB である．本研究で作成した WebFEM は，StartApplet.java，WebFEM.java，Fem_Request.java，Fem_Program.java，Fem_Model.java，Simul_Equation.java，Element.java，Node.java，Boundary_Condition.java，Material.java の 10 個のプログラムソースをコンパイルして生成されるクラスによって構成される．そのうち，WebFEM.java と Fem_Request.java は，HORB によるコンパイル時に，WebFEM.java を元にして，WebFEM_Proxy.class と WebFEM_Skeleton.class が生成され，さらに Fem_Request.java を元に，Fem_Request_Proxy.class と Fem_Request_Skeleton.class が HORB により自動的に生成される．このように HORB を使用することで，ネットワーク上のオブジェクト間の通信プログラムを自ら作成する必要が無いため，通常のプログラミングとの違いをほとんど意識する事無しに，容易に分散オブジェクト技術を使用する事が出来る．WebFEM.java と WebFEM_Proxy.java のプログラムソースの 1 部を図－4 と図―5 に示す．



図－2　ORB フレームワーク

表－1　システム開発環境

OS	Windows98 Second Edition
Java開発環境	JDK1.3.0
Webサーバ	Apache1.3.12
ORB	HORB2.0
ブラウザ	Internet Explorer5.0



図－3　システムモデル図

```
public class WebFEM {
    final String START = "<HTML><HEAD><font size=6><center>結果の表示</cente
                                                          "<HR></H
    final String END = "</CENTER></BORDER></BODY></HTML>";
    static String[] data2 = new String[1000];
    static int counter;
    static int counter2;
    static int counter3;

    public void start(String data) {
            Fem_Request_Proxy FRProxy = new Fem_Request_Proxy("horb://yb1.c
            FRProxy.start_fem(data);

            counter = FRProxy.read_Data();
            for(int i=0;i<counter;i++) {
                    data2[i] = FRProxy.send_Data(i);
            }

            counter2 = FRProxy.send_ctrldata(1);
            counter3 = FRProxy.send_ctrldata(2);

            write();
            make_html_file();
    }

    public void write() {
                try {
```

図－4　WebFEM.java のソースコード

```
public class WebFEM_Proxy implements horb.orb.Proxy {
  protected short classNo;
  public final static short major = 0;
  public final static short minor = 0;
  public short _getMajorVersion() { return major;}
  public short _getMinorVersion() { return minor;}
  protected HorbURL url;
  private transient IOCI ioci;
  public IOCI _getIOCI() { return ioci; }
  public void _setIOCI(IOCI ioci) { this.ioci = ioci; }
  public HorbURL _getObjectURL() { return url; }
  public WebFEM_Proxy() {
    classNo = 1;
  }
  public synchronized void _connect(HorbURL url, String user, String passwd) th
    try {
      if (ioci == null)
        ioci = (HORB.getFactory(url.getProtocol())).createIOCI();
    } catch (Exception e) {
      throw new HORBException("can't create IOCI: "+e);
    }
    this.url = ioci.connect(url, "WebFEM",major, minor, user, passwd, horb.orb.
  }
  public synchronized void _release() {
    ioci.releaseObject();
    ioci = null;
  }
  public WebFEM_Proxy(HorbURL url) throws HORBException {
    this(url, null, null);
```

図－5　WebFEM_Proxy.java のソースコード

5．WebFEMの使用例

本研究で開発したWebFEMを使用して，簡単な2次元静的弾性有限要素解析を実施した事例を記す．解析モデルは高さが10メートルの鉄筋コンクリート製の水槽の壁であり，形状，荷重および物性値を図－6に示す．

まず，ユーザはFEMコードへの入力データ（text file）をあらかじめプリプロセッサ等により作成しておく．Webサーバにおいては，Apache[5]及びHORBが常に動作しておりWWWに接続されている．クライアントもWWWに接続されており，ユーザはインターネットブラウザからサーバ上のURL（http://yb7.****/StartApplet.html）にアクセスする．（注：****はマシンのセキュリティのためである．）StartApplet.javaのソースコードを図－7に示す．

そしてユーザは，図－8に示すような入力画面から先の入力データを入力し，「送信」ボタンをクリックする．

するとStartAppletクラスの中の

```
data = ta.getText();
```

により，入力データが変数dataに格納され，

```
WebFEM_Proxy WebProxy =
new WebFEM_Proxy("horb://yb7.****");
```

により，WebFEM_Proxyクラスのインスタンス WebProxyが作られ，クライアントはWebProxyを経由してサーバyb7と通信出来るようになる．

次に，

```
WebProxy.start(data);
```

により，WebFEM_Proxyクラスの中のメソッドstart(data)が動作する．すると，

```
Fem_Request_Proxy FRProxy =
new Fem_Request_Proxy("horb://yb1.****");
```

により，Fem_Request_Proxyクラスのインスタンス FRProxyが作られ，このFRProxyによりサーバyb7と別のサーバyb1が通信出来るようになる．

そして，

```
FRProxy.start_fem(data);
```

により，Fem_Requestクラスの中のメソッドstart_fem(data)が動作し，

```
make_text_file(data);
```

により，ユーザから入力されたデータを本FEMコードで使用可能なように整える．

そして，Fem_Programクラスの中のメソッド

```
Fem_Program.main();
```

により，FEM解析をyb1上で開始する．その時，Fem_Programクラスでは，Fem_Modelクラスのインスタンス FMobj，Simul_Equationクラスのインスタンス SEobjを生成し，

```
FMobj.input();
```

により，入力データを読み込み，次に

```
SEobj.solve(FMobj);
```

により，FEM解析が実行され，最後に

```
FMobj.output();
```

により，解析結果をyb1上のoutput.txtに出力する．

次に，Fem_Requestのメソッド

```
read_Data();
```

により，解析結果がyb7へと送られる．

そして，test5のメソッド

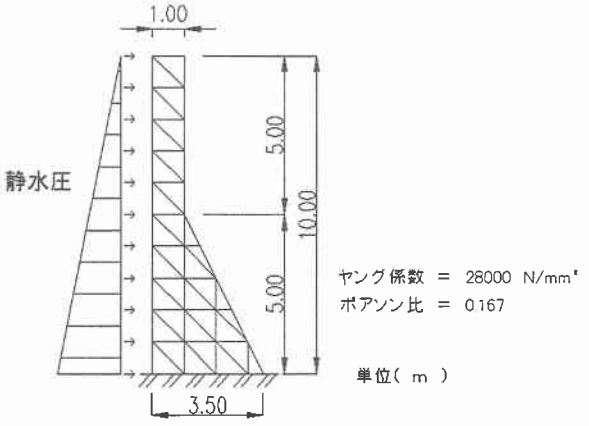
```
make_html_file();
```

により，サーバyb7上にhtmlファイルが作成される．その後，クライアントがアクセスしているWebページ上に「計算終了」というメッセージが表示されるので，図－8の「計算結果の表示」というリンクフィールドをクリックすると，計算結果が図－9（A），図－9（B）のように表示される．このファイルを適当なポストプロセッサに入力すると，図－10のような変位図や主応力図を表示させることが可能である．

6．おわりに

本研究では，FEM解析コードの開発・メンテナンス等にかかる時間・労力等のコストを低減する事，及び1台のコンピュータに負荷が集中するのを防ぐ事を目的として，ネットワーク上で処理を分散する事の可能なWebベースの有限要素解析システムの開発を行った．

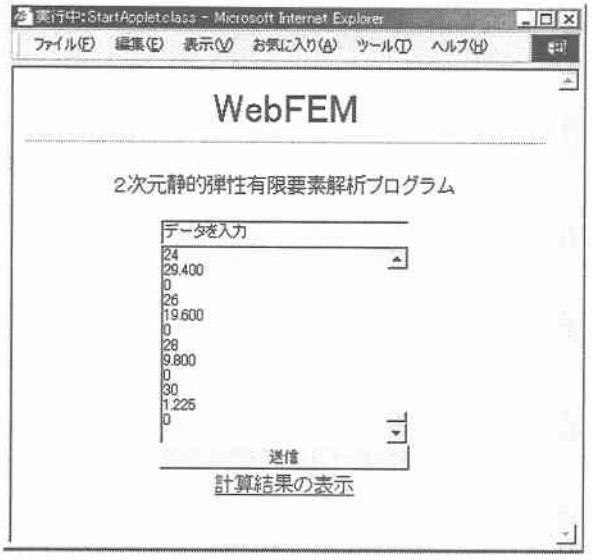
本システム開発にあたっては，まず，オブジェクト指向プログラミング言語Javaを用いてFEM解析ソフトの作成を行い，次にそのFEM解析ソフトをHORBの使用によりネットワークに対応させ，Webを通じて使用可能とした．



図－6　水槽壁のFEM解析モデル

```
public class StartApplet extends Applet {
    TextField tf;
    public static TextArea ta;
    String data;
    Button bt1;

    public void init() {
        setLayout(new BorderLayout());
        add("Center", ta = new TextArea(20,20));
        add("South", bt1 = new Button("送信"));
        add("North",tf = new TextField("",20));
        tf.setText("データを入力");

        bt1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    data = ta.getText();
                    tf.setText("計算開始");

                    WebFEM_Proxy  WebProxy = new WebFEM_Proxy("horb://yb7.
                    WebProxy.start(data);

                    tf.setText("計算終了");
                } catch(Exception e) {
                    tf.setText("例外発生(pro) : " + e );
                }
            }
        });
```

図－7　StartApplet.javaのソースコード



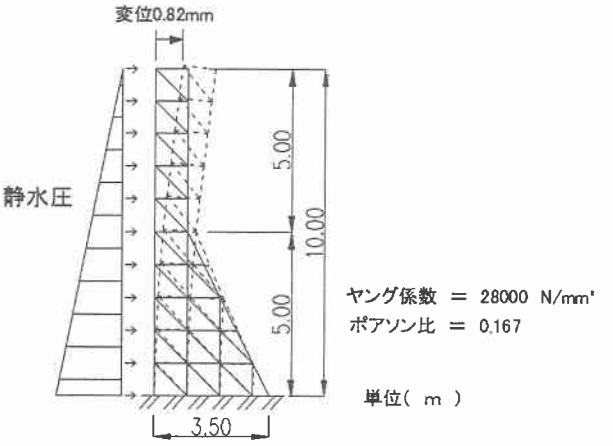
図－８　入力画面

節点番号	X方向の変位	Y方向の変位
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.021133015	0.024909446
7	0.01707556	0.006237617
8	0.016443634	-0.0072681364
9	0.016738828	-0.014644974
10	0.052787654	0.046080448
11	0.04822769	0.010750511

図－９（Ａ）　出力結果（節点変位，単位：mm）

要素番号	SX	SY	SXY
1	0.14986268	0.7475186	0.25352374
2	-0.0842346	0.16277656	-0.01914977
3	0.03752737	0.18718742	0.20484819
4	-0.06269096	-0.22191457	0.03524451
5	-0.043727282	-0.21811274	0.19726726
6	-0.07924996	-0.43771154	0.1123118
7	-0.08810854	-0.43948752	0.20080857
8	0.0056091123	0.6109192	0.15574923
9	-0.1096911	0.10799532	-0.05011939
10	0.008187238	0.13162759	0.21169603
11	-0.10486816	-0.2952469	0.031899378
12	-0.048419833	-0.28393012	0.27377847
13	-0.100988895	-0.5427301	0.17009258
14	-0.025761861	0.52663636	0.13440219

図－９（Ｂ）　出力結果（応力，単位：N/mm²）



図－１０　解析結果（変位図）

ユーザはあらかじめ解析モデルへの入力データを作成し，Web ブラウザを通じで Web サーバにアクセスすることで，FEM 解析を容易に行う事が出来る．FEM 解析を行う為に最低限必要なものは Web ブラウザのみである．ユーザ側では FEM 解析ソフトの開発・メンテナンス等を行う必要が無いことから，時間・労力等のコストを大幅に低減させることが可能だと思われる．また，こうした WebFEM を適正な課金により利用することが可能な ASP（Application Service Provider）を設立することは新たなビジネスチャンスになり得ると考えられる．

今後の課題としては，本研究で開発した FEM 解析コードをより高度化させること，Web 上で解析モデルを作成し，その解析モデルから直接解析を行うためのプリプロセッサの開発，解析結果を可視化するポストプロセッサの開発，及び並列分散処理によるさらなる高速化等が考えられる．さらに解析結果のデータベース化を行い，解析モデル・解析結果の事例を蓄えていく事で，より有効性の高いシステムとなっていくと考えられる．

参考文献

1）Peng, J., McKenna, F., Fenves, G. L., and Law, K. H.:An Open Collaborative Model for Development of Finite Element Program, Proceedings of the Eighth International Conference on Computing in Civil and Building Engineering, pp.1309-1316, 2000.

2）春海佳三郎，大槻明：有限要素法入門，共立出版，1996.

3）矢川元基，関東康祐：オブジェクト指向計算力学入門，培風館，1999.

4）http://horb.etl.go.jp/horb-j/

5）http://www.apache.org/