

WWW を用いたインターネット上での可視化システム

4E-6-1

上沢和真* 土井章男** 小山田耕二**
岩手大学工学部* 岩手県立大学ソフトウェア情報学部**

インターネット上でシミュレーション結果を公開する試みが盛んである。これまでの可視化が閉ざされた環境の中で行っていたものに対し、多人数に公開できるインターネット上での可視化が期待されている。我々はその要望に対し、JAVA 言語と VRML, そして WWW 技術を用いて可視化システムを構築した。可視化対象としては CFD シミュレーション計算結果を使用した。可視化処理については独自に作成したリモート可視化システムによりおこなった。どちらのシステムも Java アプレットを基盤としたシステムとなっており、ユーザからのパラメータ入力にも対応している。

1. はじめに

可視化技術の向上の結果、シミュレーション技術者は表現力のある手法でシミュレーション結果を可視化できるようになった。しかし、シミュレーション結果を多くの人に見せることは難しい。多人数に情報を公開する場として、インターネットが考えられる。近年のインターネットの発展により、情報を公開する技術は飛躍的に発展し、その中でも WWW 技術はパソコンなどの低機能の計算機でも情報を得ることができる技術となっている。また、性能向上によりパソコンでも複雑な 3 次元表示が可能となった。我々は WEB 上での可視化システムを 2 つの方向から考え構築した。一つはユーザ側ですべての可視化処理を行う方式で JAVA 言語で構築した。2 つめはサーバ側で可視化処理を行い、出力として VRML を出力し、クライアント側では VRML ビューワーとユーザのパラメータ処理用の JAVA アプレットで構成した。どちらの方式もユーザは自由にパラメータを設定することができ、VRML を用いたシステムでは 3 次元表示も可能となった。以下ではこれまでの可視化システムを述べ (2 章)、我々の可視化システムについて説明し (3 章)、適用例を示す (5 章)、そして最後にまとめを示す (6 章)。

2. これまでの可視化システム

ワークステーションがまだ高価だった時には、1 台の計算機上でシミュレーションから可視化処理まで一環して行うことが多かった。

ワークステーションの発展とともに X-WindowSystem を用いたクライアントサーバ方式の可視化処理が行えるようになった。この場合にはサーバで可視化処理したものをクライアントで画像または格子データ等をクライアントへ送り、処理を行い、3 次元表示できるようになった。しかし、可視化処理したものを多人数に見せる場合や、離れた場所から可視化を行う際などには使用しづらい、または使用することができないため、LAN 内からの利用にとどまっていた。

インターネットの発展とともに、WWW サーバと可視化ツールを用いたシステム構成が可能となっている。市販のツール^[1]では可視化を行い WEB 上に VRML 出力するものや、AVS を基に CGI を用いて VRML へ出力するものがある^[2]。しかし、これらで出力される VRML ファイルは、送信する VRML ファイルのサイズが大きくなってしまい、特にローエンドのクライアントでは表示が困難となってしまう可能性がある。

Visualization System On World Wide Web

Kazuma KAMISAWA

Computer and Information Science Iwate University

4-3-5 Ueda, Morioka, Iwate-ken, Japan

Akio DOI, Koji KOYAMADA

Iwate Prefectural University Faculty of Software and Information Science

152-52, Sugo, Takizawa-mura, Iwate-ken, Japan

3. 可視化システムの提案

3.1 可視化システムに求められる機能

- ① 世界中のどこからでも可視化可能であること
インターネットを利用し、WWW ブラウザと VRML ビューワーが使える環境で、比較的低機能で一般的な PC でも可視化可能であること。

- ② パラメータの変更が可能なこと
対象を様々な視点から可視化することでユーザーが理解を深められること。
- ③ 3次元表示が可能なこと
ユーザーの理解度の向上には3次元表示は欠かせない。また、低速の計算機でも可能であること。

3. 2 システム案 1 - JVS (Java Visualization System)

JVSはJAVA言語によって書かれており、JAVAコンパイラでバイトコードにコンパイルされ、WWWブラウザに搭載されているJAVA仮想コンピュータ上で実行される(図1)。

JVSは3つのオブジェクト - ボリューム、ジオメトリ、イメージから構成される。JVSが起動されるとボリュームオブジェクトはCFDプログラムのようなシミュレーション計算プログラムの結果をネットワーク上の計算サーバから読み込み、数値データから構造格子のデータ(ボリュームデータと呼ぶ)に変換する。作成されたボリュームデータからジオメトリオブジェクトは流線、等値面、断面等(ジオメトリ)を作成する。そしてジオメトリはイメージオブジェクトでレンダリングされ、イメージ化、表示される。レンダリングされたイメージはパラメータを入力することで視点等を変更できる。

JVSはメインウインドウ、可視化結果を表示するSurface Type 選択ウインドウ(図2)、視点等のパラメータ入力ダイアログ(図3)、画像表示ウインドウから構成される(図4)。

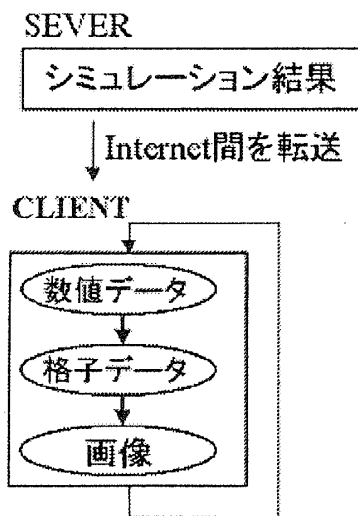


図1 JavaVisualizationSystem 全体図

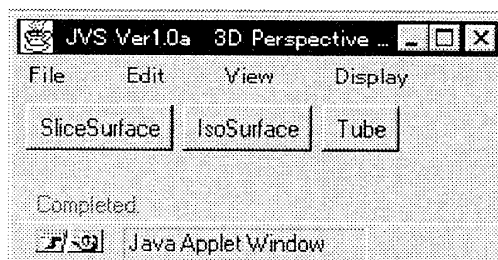


図2 メインウインド

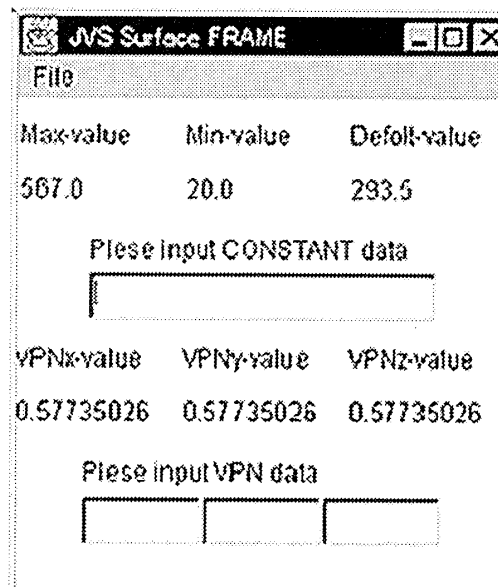


図3 パラメータ表示ウインド

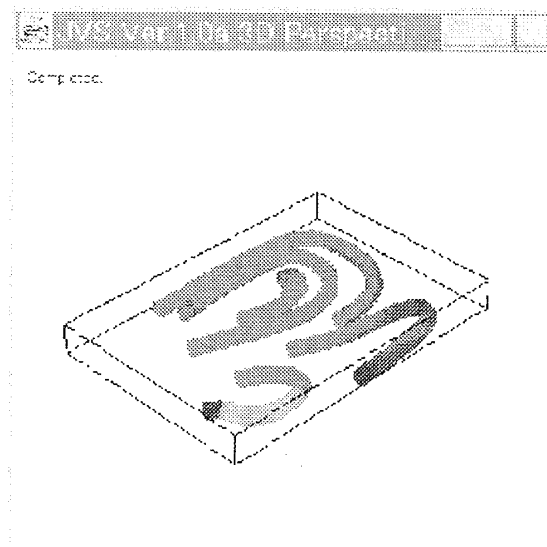


図4 画像表示ウインド

JVSはシミュレーション結果をネットワークを通して可視化することができる。計算サーバが作ったシミュレーションデータをJVSを用いることで場所を選ばず見ることが可能である。さ

らに JAVA 言語はクロスプラットフォームに対応しているため、今回 CPU も OS も異なる WindowsNT と SGI IRIX6.4 上で動かしたが問題なく動作した。このことにより、従来の機種ごとにバイナリを用意する必要がなく、バイナリはネットワークを通じてダウンロードし実行することが可能である。ユーザインタフェースについて JAVA 言語では AWT(Abstract Window Toolkit) と呼ばれる GUI ライブラリがあり、機種に依存せず同じ見た目(Look & Feel)を持つことが可能であるため有効である。

3. 3 システム案 2

VRML と JAVA 言語を使用 する形式

3. 3. 1 サーバ部

システム案 2 で大きくわけて 3 つの部から構成される (図 5)。まず、シミュレーション結果から VRML ファイルへ変更する VRML 作成サーバ、クライアントからの指示で起動、計算を指示する通信サーバ部、そして VRML を表示したり、ユーザからパラメータを取得するクライアント部である。

通信サーバは Java 言語のアプリケーションとして作成している。このサーバの機能としては次の 3 つである。

1 つめはシミュレーション結果を格納し、どのようなシミュレーションが行われ、可視化できるかをユーザへ提供する機能である。この機能はクライアント側のアプレットへシミュレーション結果の表示を行う機能である。2 つめとして VRML 作成サーバのプロセス管理である。VRML 作成サーバは C 言語で作成されているので JAVA 言語で作成された本システムからは直接管理できない。そこで一時ファイル (以下では通信ファイルと呼ぶ) を通してプロセス管理を行う。主なプロセスとしてサーバの起動、終了、ユーザからのパラメータによる VRML ファイル作成である。

3 つめはクライアントとの通信管理である。本システムはユーザからの自由なパラメータ設定により、可視化する対象データの視点等を自由に変更することが求められる。そこでクライアントとの通信には TCP-IP のソケット通信を使い対応した。

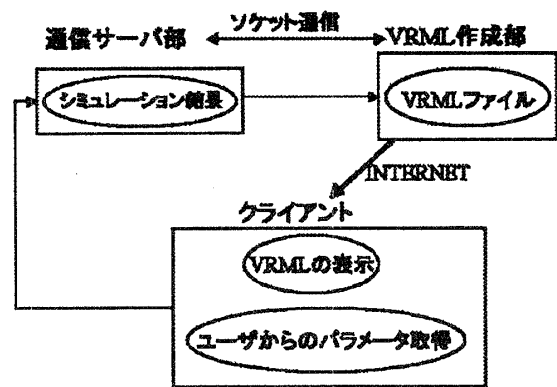


図 5 Java と VRML を用いた形式全体図

3. 3. 2 クライアント

クライアントは JAVA 言語のアプレットとして作成した。WWW ブラウザでアクセスすると通信サーバ (WWW サーバ) からアプレットがユーザへダウンロードされ実行される。(図 6) クライアントは VRML を表示するためのプラグインとユーザがパラメータを渡すための JAVA アプレットから構成される。パラメータの種類としては下記のものを用意される。

1. シミュレーションファイルの選択
2. 可視化手法の選択 (図 7, 8)
3. 通信サーバ選択

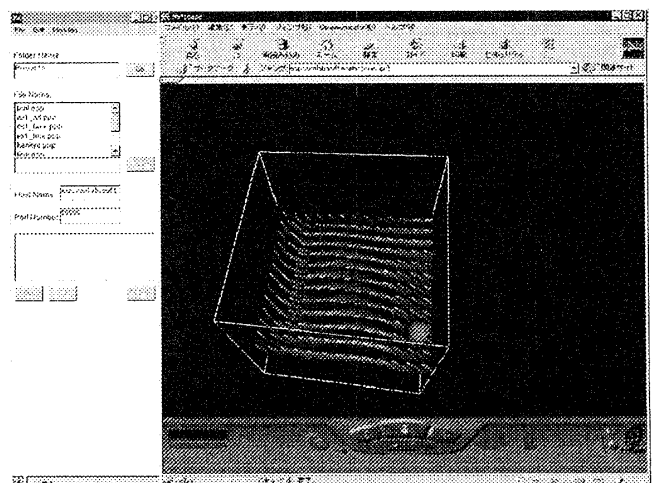


図 6 クライアント

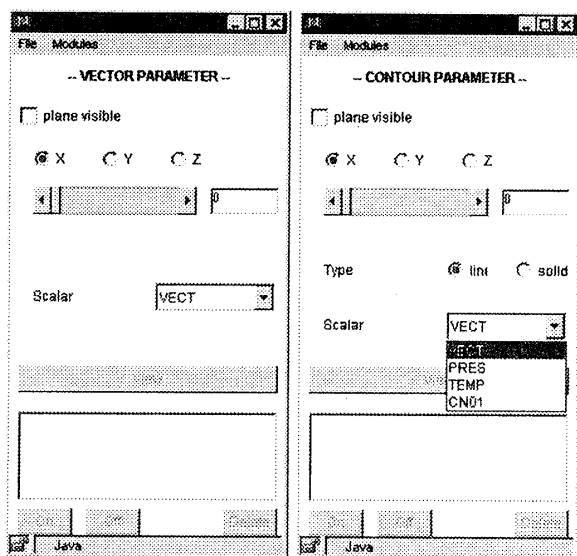


図7 選択ウインド (左:ベクタ 右:コンタ)

クライアントの主な機能はユーザからのパラメータ入力と VRML ファイルの表示である。パラメータはサーバへ転送されることで P2VRML 部を動作させ、VRML ファイルの再作成を指示することができる。VRML ファイルについてはその都度新たに作成するのではなく、ユーザからの指示により必要な部分のみの変更となる。

3. 3. 3 VRML 作成サーバ部

VRML 作成サーバ部は C 言語で作成され、シミュレーションデータから VRML へ変換する。シミュレーションデータは 3次元格子の各頂点で定義されたスカラデータ、ベクターデータとなっている。(ポリウムデータ) VRML は幾何形状を表現する言語なのでポリウムデータから幾何形状へ変更する必要がある。幾何形状を表示する手法としてあるスライス面を設定して、その面上でポリウムデータを可視化する手法を考える。ポリウムデータをスライス面にマッピングするには、スライス面と格子稜線との好転を計算し、交点を頂点とする 3 角形メッシュを作成する。頂点では、格子頂点で定義された数値データを使って補完計算を行う。スライス面上で定義される 3 角形の頂点は、スライス面と格子の稜線との交点である。その数は、格子の数の 2/3 乗に比例し、数十万の格子からだ数万个程度となる。頂点で定義されたスカラデータを元に等高線を作成するか、スキャンコンバージョンアルゴリズム [3] を使い面内の塗りつぶしを行う。

3. 3. 4 動作例

1. サーバ起動

今回、通信サーバについては WWW サーバと同一の計算機を使用する。起動するとまず、VRML 作成サーバが起動ができるか、使用できるポート番号、通信ファイルの作成チェックが行われ、クライアントからのソケット通信を待つ (図 8)。

```

bash-2.02$ cd PostServer/
bash-2.02$ /usr/J1.1.6/bin/java Server
port_number:50000
app_path:../p2vrml/NEOATR/TEST/p2vrml.exe
url_path:../PostServer
*** start listen-connection
type-get:1
currentdir-put:/home/esp/PostServer
url_path:../PostServer
port-get:50000
*** start work-connection
newport-put:1082
Thread Count:1
type-get:1
type-put:1
read_dirname:Projects
listenrun socet.close()
workrun socket.close()
    
```

図 8 サーバ動作画面

2. クライアント起動 (アプレット)

アプレットは WWW ブラウザからダウンロードされ、実行される。(図 6) 起動とともに、指定されたサーバへアクセスし、使用できる TCP-IP ポート番号を取得する。(TCP-IP ポート番号はクライアントごとに指定される) ポート番号がきまると可視化可能シミュレーションファイル名一覧を取得する (図 9)。

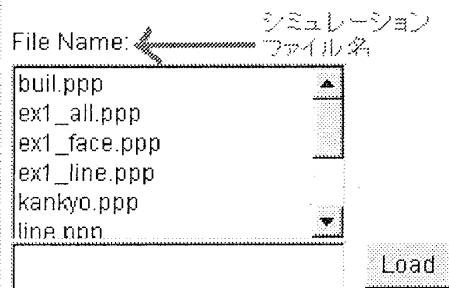


図 9 シミュレーションファイル名取得

3. 初期条件での VRML の表示

シミュレーションファイルにはそれぞれ初期条件がつけられており、それによって VRML 作成サーバ

が VRML ファイルを作成する。作成された VRML ファイルはクライアントに送られ表示される（図 1 0）。

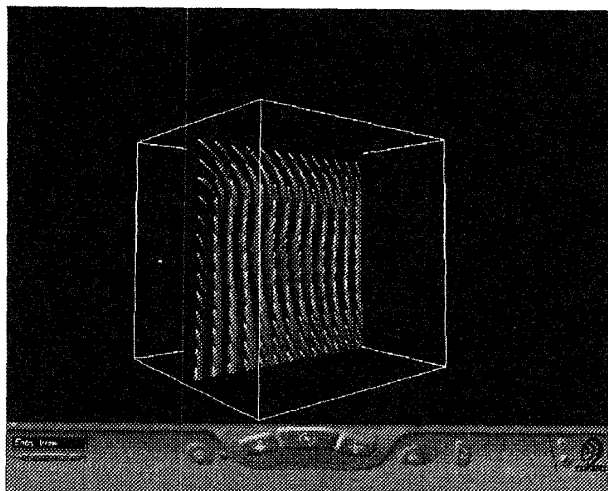


図 1 0 シミュレーションの表示例

4. ユーザからの VRML 作成パラメータの入力
 初期条件とは異なるパラメータを使用したい場合には図 4, 5 のようなパラメータ入力ウインドーに入力する。パラメータはサーバに送られ VRML 作成サーバによって VRML ファイルが作成され、ユーザに再度送られる。

4 適用例
 4. 1 JVS

1. 等値面の表示例（図 1 1）
 ポリゴン数 860個
 実行速度（可視化処理分） 20秒

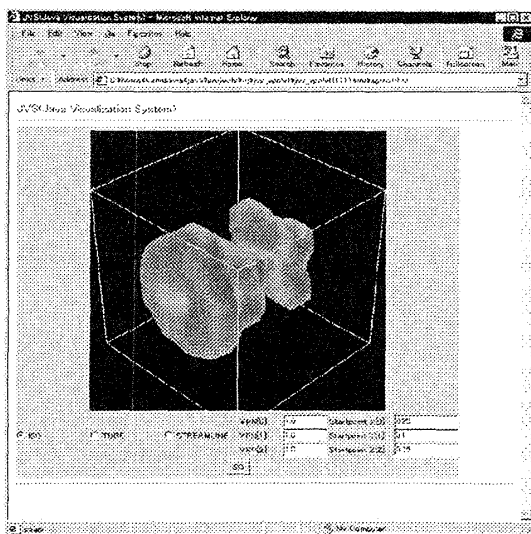


図 1 1 等値面表示

2. スカラー場の表示例（図 1 2）
 ポリゴン数 80個
 実行速度（可視化処理分） 6秒

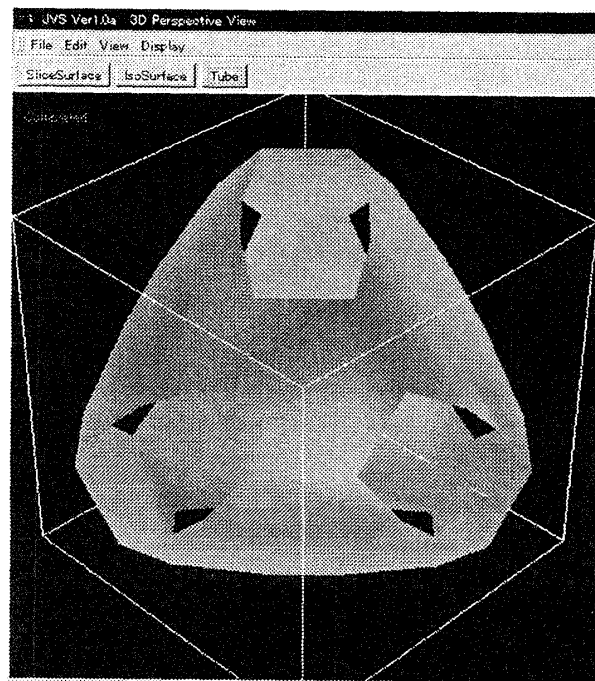


図 1 2 スカラー表示

4. 2 Java と VRML を用いた形式

1. ベクトルを矢印表示したもので、渦を表示している。赤い球は渦の中心へアニメーションをする。（図 1 3）

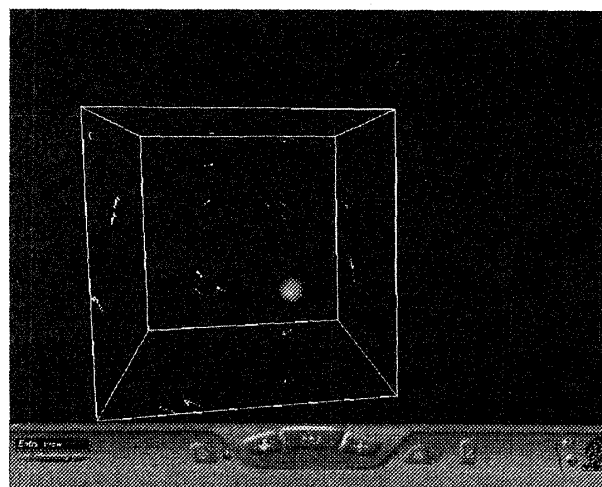


図 1 3 渦の中心への矢印表示

2. スライス断面でのフランジ表示 (図14)

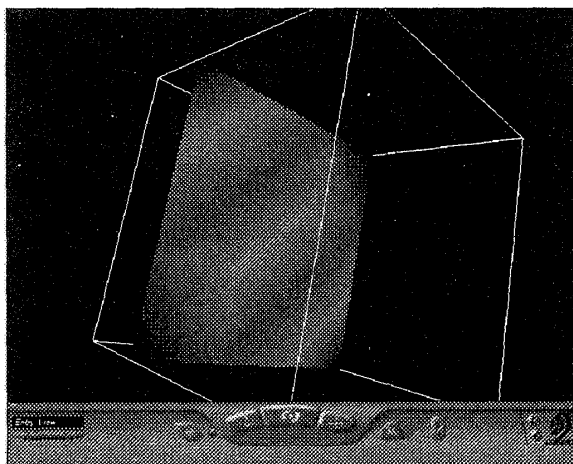


図14 スライスの表示

3. 拡散物質とともなって発生する渦の表示 (図15)

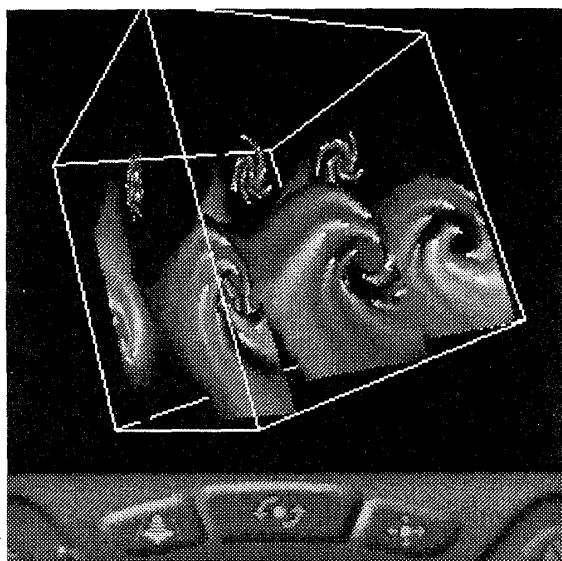


図15 渦の表示

6. 総括

6.1 可視化処理の速度

JVS についてはサーバからシミュレーション結果を受け取り、可視化処理自体はアプレットのみで行う。そのため、ネットワークの速度が遅い、シミュレーション結果を格納するサーバから遠いなどという場合、可視化対象を変更することがなれば、一度のファイル転送で済むため、パラメータの変更が多い場合には有効である。しかし、

シミュレーションの対象をなんども変更するような場合にはネットワークの速度に依存してしまう。

一方、VRML を用いたシステムの場合には可視化する対象が巨大であっても可視化処理を行うサーバで VRML ファイルにする過程でポリゴン数の削減などが行われるため、パラメータの変更が少ない場合などは有効であるが、パラメータの変更が多い時にはネットワークの速度に依存してしまう。

6.2 インターフェース

JVS と VRML を用いたシステムとの一番の違いは JVS が計算は 3 次元で行うが表示は 2 次元であるのに対して VRML を用いるシステムの場合には 3 次元表示ができる点である。JVS の場合にも視点等の変更の場合にはパラメータを入力すればすぐに画像に反映されるが VRML のようにマウスで自由に変更することはできない。

以上の点からおのこのシステムで問題点はあるが WWW を用いた可視化システムとして動作することができた。今後の課題として以下の点が上げられる。

1. ユーザによるシミュレーション

現システムにおいてユーザは、シミュレーション結果の可視化のみを行っている。今後はシミュレーション自体もユーザ側から指示を出せる改良が必要である。

2. VRML 以外の 3 次元表示

JVS において今後 JAVA 3D などを用いて 3 次元空間を表現できる改良が必要である。

3. パラメータの増加。

現在、ユーザが入力できるパラメータはシミュレーションファイル名、ベクタ場、等値面に限られている。これを、今後増やしてユーザに提供するサービスを増やす必要がある。

参考文献

- [1] IRIS Explorer 日本 NAG
<http://www.nag-j.co.jp/ie4.html>
- [2] 宮地英生, 中野誠己: AVS を使った可視化サーバの試用公開, ながれ, Vol16, pp316-317(1997)
- [3] 小山田耕二: VRML による可視化のためのデータ削減技術, ながれ, Vol16, pp323-327(1997)